Alfonso Mur / Pablo Nieto / Jesus Molina

Virus informáticos

BIBLIOTECA INFORMATICA



Es una promoción de:



AntiVirus Anyware

Virus informáticos

Virus informáticos

Alfonso Mur
Pablo Nieto/Jesús Molina
Diplomados en Informática de Gestión
por la Universidad Pontificia de Comillas

BIBLIOTECA INFORMATICA

Diseño de cubierta: Vital García

Impresión: C.G.A

ISBN: 84-88337-09-4
Depósito Legal: 28951-94
Impreso en España
Printed in Spain, 1994

Ejemplar gratuito. Prohibida su venta.

A todas las personas
que con su ánimo, calor y comprensión
han ayudado en este libro.

Indice

Apéndices

Prólogo

"Ya nada volverá a ser como antes. Hemos perdido nuestra inocencia informática. Se acabaron aquellos alegres días en los que usábamos sin preocuparnos *diskettes* de programas que nos suministraban los amigos o que obteníamos de cualquier fuente y que tal vez eran copias no autorizadas. De repente los virus informáticos irrumpen en nuestra vida y nos hacen replantear nuestros hábitos de conducta como usuarios finales."

Las noticias de grandes catástrofes empiezan su loca espiral envolvente y en ese ojo del huracán nos encontramos inmovilizados, sin saber qué hacer ante la posibilidad o la realidad de que nuestra biblioteca de programas en *diskettes* o en disco duro esté contaminada.

¿Qué hacer? ¿Qué determinación tomar cuando nuestros programas no se comportan de la forma habitual? ¿Qué acciones preventivas realizar para evitar que los virus informáticos contaminen nuestros programas?

No hay que alarmarse, la situación tiene arreglo, y la mejor manera para ello es estar informado. Los autores de esta guía, gracias a su conocimiento de la casuística de los virus informáticos y a una ingente labor de recopilación, nos presentan de una forma amena, profunda pero clara, la información y las respuestas a las preguntas que todos nos hacemos sobre este fenómeno de contaminación informática.

A través de la lectura de esta guía conoceremos qué es un virus, cómo se origina y se desarrolla, qué tipos de virus hay, medios de contagio y formas de evitarlo. Asi-

mismo, tendremos una visión de las disposiciones legales tanto en Estados Unidos como en España. Por último, encontraremos una serie de apéndices muy útiles para el lector que desee una mayor información técnica sobre diversos aspectos de los virus informáticos.

ROMAN HUERTAS GARCIA
Ingeniero Superior de Telecomunicación
Licenciado en Informática

Introducción

En los dos últimos años han aparecido innumerables artículos en todo tipo de prensa referentes a este fenómeno informático. Cada uno ofrecía su propia definición, más o menos acertada, de lo que es un *virus informático*. Esta variedad de opiniones ha creado una atmósfera de confusión.

Titulares como "Un virus comparable al SIDA afecta a los ordenadores de todo el mundo", "Virus informático: una epidemia galopante", "Los ordenadores de EE.UU. están afectados por el SIDA tecnológico", "Un virus informático anda suelto" o "El SIDA informático ya está en España", recogidos, tanto en prensa de renombre como en publicaciones técnicas de nuestro país, unidos a la deplorable información ofrecida en los medios audiovisuales, han llevado a la opinión pública no especializada a realizar comentarios de todo tipo sobre el suceso informático de los ochenta.

No se pretende con esta guía ofrecer una definición universal de virus informático, sino simplemente explicar de la forma más asequible para el usuario, qué es un virus, cómo se extiende, cómo afecta a los ordenadores, programas y datos, y qué pasos se pueden seguir para la prevención del fenómeno.

Los autores desean agradecer a todas las personas que de forma directa o indirecta han ayudado a que este trabajo viera la luz, a aquellas que con su cariño, comprensión y estímulo han animado a seguir en los momentos más difíciles.

En particular, quieren expresar su gratitud a Fernando

Cuéllar, por el interés que se ha tomado en que se plasmara la idea por escrito, y a la gente del Departamento de Prensa de Hispania Service, sin cuya colaboración el trabajo hubiera quedado incompleto. A todos, muchas gracias.

Cómo usar este libro

La proliferación continua de virus en el mundo informático despertó, desde las primeras noticias, un interés especial para los autores del libro. Cuidadosamente se fue seleccionando toda la información disponible en los distintos medios de comunicación audiovisual y en la prensa nacional e internacional, tanto si se trataba de carácter general como de información técnica, consiguiendo un voluminoso archivo.

El objetivo que persigue este trabajo es aumentar el conocimiento sobre el fenómeno informático de la década, en todas aquellas personas que por dedicación u obligación se ven frente a la pantalla de un ordenador. No se pretende que sea la panacea contra el virus informático, sino una guía práctica que oriente al usuario de ordenadores sobre el peligro que le acecha, y derribe el halo de mito que ha venido envolviendo a este tipo de programas.

En esta guía se podrá encontrar una definición exhaustiva del concepto de "virus" y una descripción de los diferentes tipos que existen, de acuerdo con una división basada en su forma de propagación. También se recoge una breve historia de los grandes casos que han afectado a las redes mundiales de información, así como una serie de medidas de prevención, detección, protección y contingencia para entablar una lucha contra la plaga informática.

En los apéndices incluidos se ofrece al lector un repaso de las nociones básicas del sistema operativo DOS, una lista del servicio de interrupciones del mismo sistema

operativo y un programa de protección contra escritura del disco duro en ensamblador. Asimismo, se hace un análisis profundo de algunas rutinas del virus Viernes-13, del que por ética profesional no se ha incluido el listado completo. Por último, se recoge una relación, necesariamente inconclusa, de los virus conocidos hasta el momento.

1

Generalidades

1.1. ¿Qué es un virus?

Con el término "virus" se designa un programa de ordenador, generalmente anónimo, que lleva a cabo acciones que resultan nocivas para el sistema informático y cuyo funcionamiento queda definido por las propiedades que se indican a continuación.

Es capaz de generar copias de sí mismo de forma homogénea o en partes discretas, en un fichero, disco u ordenador distinto al que ocupa.

Modifica los programas ejecutables a los que se adosa, o entre cuyas instrucciones de código se introduce, consiguiendo así una ejecución parasitaria. Esto implica que se pueda activar de forma involuntaria por el usuario cuando éste ejecute el programa que lo porta. Los programas portadores pueden ser de uso común del usuario o programas ejecutables del sistema operativo, siendo estos últimos el objetivo esencial del virus.

El efecto que produce un virus puede comprender acciones tales como un simple mensaje en la pantalla, la ralentización de la velocidad de proceso del ordenador o el formateo de una unidad de disco, pero no se debe olvidar que su funcionamiento intrínseco coincide con el de cualquier programa ejecutable. Esta característica supone que para que un programa de este tipo ejerza sus acciones nocivas es necesario que se active, es decir, que el código que lo conforma se ejecute. Por otro lado, debe permanecer en memoria para poder obtener así el control

permanente de la unidad central de proceso (CPU), centro neurálgico del ordenador.

Generalmente, su funcionamiento comprende dos fases bien diferenciadas. Durante un período el programa permanece oculto al usuario, en espera de una acción, como la introducción de una cadena especial de caracteres por el teclado, una fecha determinada o un tope de autocopias del virus almacenado en un contador interno. En esta fase el programa realiza una acción de esparcimiento cuyo objetivo fundamental consiste en realizar el mayor número posible de copias de sí mismo en otros soportes distintos o en el mismo que él ocupa.

Una vez que se produce este hecho, el virus realiza la acción nociva para la que ha sido programado, completando así la segunda fase de funcionamiento.

Por último, cabe destacar que un virus se diseña intentando disfrazar su presencia ante el sistema y ante el usuario. Generalmente no es descubierto hasta que, en la segunda fase del ciclo de funcionamiento del programa, surge un hecho anormal derivado de su ejecución que da la señal de alarma.

A la vista de la presente definición se pueden realizar algunos comentarios adicionales.

La denominación de virus informático corresponde a una metáfora que asocia este tipo de programas con sus homónimos biológicos. Ciertos autores han querido en encontrar en las características de los virus biológicos ciertas similitudes con los programas de sabotaje. Así ha nacido una nueva jerga informática que incluye términos como "epidemia", "contagio", "infección", "vacuna", "antídoto", "tiempo de incubación", etc.

Ciertamente podría existir una similitud en lo que se refiere al fenómeno autorreproductor del virus biológico con su metáfora informática. Incluso sería aceptable comparar el tiempo de latencia o incubación de un germen con la espera ante un hecho que desencadene el virus informático. Pero parece excesivo buscar comparaciones entre el ADN de las células atacado por un agente patógeno y la forma en que un programa nocivo se adhiere o introduce en un programa ejecutable de ordenador. Resulta también exagerado establecer correspondencias entre este fenómeno y el síndrome de inmunodeficiencia adquirida (SIDA).

Aunque resulte simple la aclaración, se hace necesario destacar que el "contagio" del virus se realiza de forma lógica a través de operaciones de entrada y salida sobre soportes magnéticos, estando el programa contaminante en ejecución y residente en memoria. Este comportamiento deshace toda hipótesis de contagio por contacto directo de los soportes magnéticos con el ordenador y, por supuesto, con las personas que lo manejan.

No se debe olvidar que el origen de todo virus es un programa "padre" desarrollado por un programador experto, encargado de iniciar la epidemia de acciones perjudiciales. Este punto es fundamental, ya que puede suponer la única prueba fehaciente cuando se trata de ejercer acciones legales contra el responsable de una "infección vírica".

Algunos autores determinan una clasificación de los programas víricos en función del efecto de las acciones que realizan. Así, han dividido estos programas en benignos y malignos. En la subdivisión de los benignos se incluyen programas que no ejercen acciones destructivas sobre la información almacenada en el soporte magnético. Un ejemplo sencillo de acción "benigna" sería un mensaje por pantalla felicitando el año nuevo.

Si se analiza un poco más en profundidad el efecto de un virus, se deducirá fácilmente que este factor de benignidad no existe. Un programa de este tipo conlleva una ejecución residente en memoria que supone una reducción del espacio operativo de la misma. Por otro lado, necesita del manejo de interrupciones del sistema operativo para sacar el mensaje por la pantalla e influye sobre el mecanismo de control de la CPU. La suma de estas acciones puede suponer una reducción patente de la velocidad de proceso en el *hardware* y un aumento considerable en el tiempo de respuesta del ordenador, lo que significa una pérdida considerable de tiempo.

Imaginemos un centro de proceso de datos de un gran banco. Al terminar el día, se programa el ordenador para que actualice las cuentas bancarias con las miles de operaciones que se han efectuado. Una tarea así se realiza durante la noche mediante un proceso *batch* o proceso por lotes. Este sistema consiste en preparar una serie de trabajos, en nuestro caso de actualización de ficheros, para que se realicen sin un control continuo del operario. El

método supone un ahorro de tiempo y un mejor aprovechamiento del sistema.

Continuemos dejando volar la imaginación y supongamos que se ha contaminado el programa de actualización con un virus "benigno" que le da las buenas noches al operador, solicitando después que se pulse una tecla para continuar el proceso normal. El programa de actualización, portador del código contaminante, comenzará su ejecución y al alcanzar dicho código, detendrá el proceso de actualización y sacará por pantalla el simpático mensaje. El proceso se mantendrá detenido en espera de que alguien pulse una tecla para continuar con la ejecución del programa de actualización. Si el administrador del sistema está pendiente de la ejecución, poco probable si la fechoría se realiza de madrugada, no supondrá una pérdida notable de tiempo. Pero ¿qué sucede si el administrador del sistema no se percata hasta pasadas varias horas o no está presente en el centro de proceso de datos? No sería complicado evaluar las consecuencias económicas que esto supondría.

La moraleja aplicable es que no hay virus benigno. Cualquier funcionamiento anómalo de un sistema de información supone un perjuicio para el usuario que implica una pérdida importante de tiempo y, por tanto, de dinero.

1.2. Origen de los virus

Los articulistas más escrupulosos han pretendido otorgar a John von Newmann la paternidad de los virus. Von Newmann, matemático brillante, hizo importantes contribuciones a la física cuántica, la lógica, la meteorología y la teoría de ordenadores. En un artículo titulado "Theory and Organization of Complicated Automata", de 1949, Newmann expone la idea de una porción de código que se reproduce y por tanto está "vivo". Años más tarde, en 1955, en su obra *The Computer and the Brain* (publicado en español en 1980 con el título *El ordenador y el cerebro*), hace una disertación teórica sobre la posibilidad de crear un autómata capaz de reproducirse a sí mismo.

Estos primeros indicios teóricos de autorreproducción del *software* y del *hardware* no parecen ser lo bastante sóli-

dos para establecer el origen. Sería algo parecido a asignar a Julio Verne la paternidad del submarino por su obra *Veinte mil leguas de viaje submarino* o la del cohete lunar por su obra *De la Tierra a la Luna.*

La clave en el origen de la difusión del fenómeno vírico se ha querido situar en una serie de artículos publicados en la revista americana *Scientific American,* firmados por A. K. Dewdney. El primero de la serie, fechado en mayo de 1984 y traducido en *Investigación y Ciencia,* versión castellana de la revista americana, en julio de 1984, se tituló "Juegos de ordenador: en el juego de la «Guerra Nuclear» dos programas hostiles entablan, sin ayuda externa, batallas de bits"*.

En este primer artículo, Dewdney explica el programa llamado Guerra Nuclear, juego en el que no intervienen activamente los usuarios. En él, dos programas hostiles se enzarzan en una lucha para obtener el control de la memoria atacando abiertamente al contrario. Estos dos programas se ponen en marcha mediante un programa ejecutor llamado MARS (*Memory Array Redcode Simulator*), que va ejecutando alternativamente las instrucciones de que constan los programas de combate, una instrucción de cada programa, de modo similar a un sistema caracterizado por compartir tiempos. Ayudado por David Jones, alumno de su departamento, va desarrollando programas cada vez mejor equipados para destruir a su contrario. Una de las versiones, denominada Gemini, tenía como única función producir una copia de sí mismo cien unidades de memoria más allá de su posición actual, transfiriendo después el control a la nueva copia.

Dewdney comenta en este artículo que el origen de su Guerra Nuclear está en una tecnología de construcción de memorias. Cita el autor dos precedentes de su juego. Por un lado, M. Douglas McIlroy, de los laboratorios americanos Telephon & Telegraph, diseñó un programa llamado Darwin. En él, cada jugador presenta cierto número de programas en lenguaje ensamblador llamados "organismos", que habitan conjuntamente en la memoria central

* El nombre original del juego es Core Wars, cuya traducción técnica es "Guerras de toros de ferrita"; la expresión hace referencia a los núcleos de ferrita en forma de toroides que componían en un principio las memorias de los ordenadores. No se entiende el origen del término "nuclear", que seguramente será una desfiguración del término "núcleo".

con los organismos presentados por los demás contendientes. Los programas creados por cada jugador, pertenecientes a una misma especie, tratan de aniquilar a los de otra especie. Gana la partida el jugador con más organismos al acabar el tiempo de combate.

Por otro lado, John F. Scoch, del Centro de Investigación de Xerox en Palo Alto, Estados Unidos, creó Worm (gusano). Este programa experimental fue ideado para obtener el máximo rendimiento de una red de miniordenadores interconectados de Xerox. Un programa supervisor se encargaba de cargar el "gusano" en máquinas inactivas para asumir el control de la máquina y, en combinación con otros gusanos residentes en otras máquinas inactivas, hacer funcionar grandes programas de aplicación en el sistema multiprocesador resultante.

En el texto de este primer artículo, Dewdney insta a los lectores a que reflejen sus ideas de programas autoprotectores y autorreparadores y establece las normas y reglas del juego. Jamás hubiera imaginado el autor lo que acontecería.

En el transcurso del mismo año se define por primera vez, de forma pública, el término "virus de ordenador". Durante la conferencia IFIC/SEC'84, en septiembre de 1984, el doctor Fred Cohen, en su exposición de la ponencia "Computer Viruses: Theory and Experiments", explica este tipo de programas como *software* maligno capaz de reproducirse a sí mismo.

El segundo artículo de Dewdney en *Scientific American*, en marzo de 1985 (mayo de 1985 en *Investigación y Ciencia*), se titula "Juegos de ordenador: virus, gusanos y otras plagas de la Guerra Nuclear atentan contra la memoria de los ordenadores". En él Dewdney pone de manifiesto las consecuencias que puede acarrear su juego gracias a los testimonios escritos de sus lectores.

Comenta el autor: "... Cuando en julio del año pasado apareció el artículo dedicado a la «Guerra Nuclear», no se me ocurrió que pudiera estar tocando un tema tan serio. Las descripciones de programas escritos en lenguaje máquina que entonces di, capaces de desplazarse de uno a otro lugar de la memoria, al acecho, dispuestos a aniquilarse el uno al otro, pulsaron una cuerda resonante". Continúa diciendo el autor: "... según muchos lectores cuyas historias y anécdotas referiré, existen multitud de

gusanos, virus y otros organismos «programáticos», que moran en todo ambiente informático concebible. Tan horripilantes son algunas de las posibilidades, que dudo si transcribirlas".

No debe interpretar el lector de está guía que fue A. K. Dewdney el inventor de los virus, más bien se debe ver como un impulsor involuntario de este tipo de programas al difundir un inocente y creativo juego. No se trata de debatir la trasparencia informativa del fenómeno, lo que no cabe duda es que el autor colaboró inocentemente en dar a conocer el fenómeno vírico cuando decidió transcribir alguna de las "horripilantes posibilidades". Estas son algunas de ellas.

Jim Hauser y William R. Buckley, de la Universidad Politécnica de California, crearon el Apple Worm, gusano de los ordenadores de la marca Apple. Este programa sacaba copias de sí mismo en un viaje a través de la memoria del Apple II con un procesador 6502.

Otra de las plagas informáticas fue concebida por Roberto Cerruti y Marco Morocutti en la ciudad de Brescia. Los dos italianos buscaron el medio de infectar el Apple II, pero no con un gusano sino con un virus. Las conclusiones que sacaron fueron que el programa tendría que infectar los discos y utilizar los ordenadores como medio de transmisión de un disco a otro. El virus era una alteración del sistema operativo contenido en cada *diskette* del Apple. Comenta Dewdney la intención, no llevada a cabo, de los programadores latinos de convertir el virus en maligno. La ocurrencia consistía en que al cabo de dieciséis ciclos autorreproductores contados en el disco contagiado, el programa decidiera reinicializar el disco inmediatamente después del primer arranque. Incluso se les ocurrió colocar sus virus en los discos utilizados en la principal tienda de informática de su ciudad. La razón triunfó y no llevaron adelante su idea.

En el artículo de *Scientific American* de enero de 1987 (*Investigación y Ciencia*, en marzo del mismo año), Dewdney abandona completamente cualquier relación de su juego con los virus y se dedica a narrar el primer campeonato del polémico juego Guerra Nuclear.

Ante la actualidad y notoriedad del fenómeno vírico y el rosario de informaciones que surgían al respecto, Dewdney retoma el tema en un nuevo artículo en *Scien-*

tific American, en marzo de 1989 (*Investigación y Ciencia*, en abril del mismo año). Bajo el título "Juegos de ordenador: sobre gusanos, virus y Guerra Nuclear", el autor se defiende de las reiteradas insinuaciones sobre la presunta relación existente entre Guerra Nuclear y los programas víricos. Justifica Dewdney su postura de transparencia informativa ante el fenómeno como un factor constructivo en el sentido de estimular los esfuerzos para la protección de sistemas. Recrimina a la prensa sensacionalista que por medio de artículos incompletos y distorsionados, escritos por columnistas que desconocen el funcionamiento interno del ordenador, ha conducido al desconcierto. Utiliza además otro argumento de peso. Las descripciones de un virus, incluso las más detalladas, no pueden utilizarse en la reconstrucción de un programa nocivo, excepto por un experto. Una persona con tal nivel de conocimientos no necesita de la lectura de revistas para crear un código que destruya los programas y los datos de otros.

Después de sentar estas bases y sin encontrar un motivo para no hacerlo, Dewdney realiza una detallada explicación del funcionamiento de los programas víricos. Se manifiesta de acuerdo con las teorías de Cohen referentes a la imposibilidad de construir un programa que detecte todo tipo de virus. Concluye este último artículo de la serie con una frase que deja claro la involuntariedad e inocencia del autor en la difusión del fenómeno vírico: "Escribir y ejecutar un virus no es la obra de un profesional de la informática y sí es la de un vándalo del ordenador. Permitamos que aquellos que pudieran contemplar actos similares prueben en vez de eso a participar en la Guerra Nuclear".

Algunas hipótesis, sin confirmación, apuntan a los grandes fabricantes de *software* como iniciadores de la corriente vírica. La justificación se establece como mecanismo de protección contra la copia para evitar que sus productos fuesen multitudinariamente plagiados sin obtener beneficios. Es evidente el quebranto económico que este otro tipo de delito produce en la industria del *software*, pero parece descabellada la idea de combatir el fuego con fuego. Este tipo de guerra acabaría perjudicando a los mismos fabricantes, que tendrían que reemplazar miles de copias de sus productos. No parece sensato que

quieran tirar piedras contra su propio tejado. Además, la tendencia actual en el *software* está acabando con las protecciones. La mayoría de las casas comerciales está desprotegiendo sus productos contra la copia. Esta justificación se refiere a los grandes diseñadores de programas, no descartándose que algún pequeño fabricante haya utilizado estas técnicas de protección.

Otras versiones juegan con la paradoja del huevo y la gallina. ¿Qué fue primero, el virus o el antídoto? Insinúan que los virus proceden de fabricantes de programas que los combaten. Aunque la idea no parece del todo descabellada, no resulta demasiado sólida, dada la gran cantidad y variedad de programas contaminantes que circulan en los ámbitos informáticos.

Resulta realmente complicado establecer la verdadera procedencia de este tipo de programas. Seguramente no surgieran de forma aislada en un único lugar y como idea de un único programador, sino que son el fruto de la ocurrencia simultánea de distintos programadores malévolos en varios países.

1.3. ¿Cómo y por qué se desarrolla el virus?

Cuando la ciencia de los ordenadores empezaba a dar sus primeros pasos, el objetivo primordial consistía en procesar información de la manera más rápida en que el hombre era capaz de hacerlo. Los factores como capacidad de conexión, portabilidad o telecomunicación parecían extraídos de novela futurista.

Los sucesivos peldaños por las distintas generaciones de ordenadores, desde los relés y tubos de vacío a los microprocesadores, pasando por transistores y circuitos integrados, fueron asegurando los conceptos de comunicación.

El auge del ordenador personal, la utilización de la línea telefónica para la transmisión de datos codificados a través del *modem*, las redes de área local (LAN), las redes públicas de ordenadores y las comunicaciones vía satélite, unidas a la estandarización de sistemas operativos, protocolos de comunicación y lenguajes de programación

han desembocado en una única tendencia informática: la posibilidad de conexión.

La capacidad de conexión se define como la característica que permite la conexión entre ordenadores, abarcando factores como la compatibilidad del *software*, los formatos estándar de información, líneas telefónicas y transmisiones por satélite. Estos factores se engloban en la transmisión remota de datos, denominada telecomunicación.

Hoy día no se concibe un ordenador cuyas principales características no sean la capacidad de conexión del *hardware* y la portabilidad del *software*. Todo ello con un único objetivo, el mayor grado de comunicación. Esta característica es un reflejo del mundo laboral, que evoluciona de manera imparable hacia el manejo de la información. Quien tiene la información tiene el poder.

La tendencia de facilitar al máximo las "relaciones" entre ordenadores constituye, sin embargo, el mar encalmado en que navegan placenteramente los piratas informáticos. Como se resaltaba en la definición de virus, la principal característica de este tipo de *software* es la autorreplicación o autocopia. Surge, pues, el pez que se muerde la cola. Se tiende a favorecer la interconexión de ordenadores ya sea por vía física o por vía lógica. Por otro lado, este factor de comunicación y capacidad de conexión es el medio más propicio para la proliferación del *software* pernicioso.

No tiene mucho sentido pensar que evitando la comunicación informática se erradicarían los virus; no es muy razonable matar al mensajero. Por ello, la solución para asegurar buenas comunicaciones sin peligro de contagios consiste en dirigir los esfuerzos hacia la seguridad de los sistemas. Se hace necesario un buen plan informático que asegure la fiabilidad de las comunicaciones.

La comunicación como vía de transmisión del virus refleja algún aspecto más que la conexión física entre ordenadores. El segundo argumento de peso en la proliferación del fenómeno se encuentra en el "arte" de copiar programas originales, en la piratería del *software*. Es patente la reducción constante en el precio del *hardware* en el mercado internacional. La alta competencia está obligando a los fabricantes de *hardware* a modificar a la baja sus precios, permitiéndose únicamente algunas frivolidades económicas en los últimos adelantos tecnológicos.

Como consecuencia, las ventas se han disparado y el parque informático, concretamente el segmento de ordenadores personales, aumenta. Este factor está haciendo que el componente mecánico de la informática sea fácilmente asequible para todo tipo de usuarios.

No sucede lo mismo con el componente lógico. La especialización de las aplicaciones exige que el usuario posea una buena colección de programas para sacar rendimiento a su ordenador. Los fabricantes de *software* se resisten a entrar en una guerra de precios y mantienen unos precios francamente elevados. Por este motivo prolifera una "industria" sumergida dedicada a la copia ilegal de programas. Este es otro foco importante de expansión de los virus.

Otra vía de entrada de programas contaminados la constituyen dos tipos especiales de *software*. En la adaptación del lenguaje informático se han divulgado infinidad de términos sajones. Dos de ellos, pertenecientes a la rama de los acabados en "ware", son *freeware* y *shareware*. Estas dos modalidades de *software* hacen referencia a aquellos programas obtenidos sin pagar dinero. Son de difusión pública y se consiguen en los BBS (*Bulletin Board Systems*), que se podrían definir como clubs de intercambio de *software* por vía telefónica.

Los programas *freeware* son gratuitos, mientras que los programas *shareware* se obtienen pagando una módica cantidad en concepto de derechos de autor o bien a cambio de la documentación de dicho programa. El uso común de este tipo de *software* lo ha llevado a convertirse en una fuente importante de difusión de virus. En la actualidad, los administradores de los BBS se esfuerzan en mantener sus clubs de intercambio sin contagio alguno.

Al explicar el motivo de este delito informático, surgen los mismos problemas que al buscar su origen. Saldría del contexto de esta guía establecer un perfil psicológico del creador de un virus, pero el sentido común lleva a pensar que el móvil puede ser de tipo económico, para sacar provecho de un posible antídoto, o bien para obtener una falsa notoriedad en el acto de haber iniciado una plaga informática. Queda patente que el creador de un programa de estas características no se detiene a evaluar el perjuicio que su programa pueda acarrear.

1.4. Impacto y alcance: Transparencia informativa

Antes de calibrar la influencia del virus en el mundo de la informática, merece la pena detenerse en el conflicto existente entre el silencio y la transparencia informativa del fenómeno.

Entre los periodistas ha existido siempre una división de posturas cuando se trata de abordar ciertos temas delicados. En nuestro país ha sido criticada la transparencia de los medios de comunicación en relación al terrorismo político. Argumentaciones de peso, como la publicidad que se da a las bandas terroristas con la información de los actos que cometen, se esgrimen para rebatir la demanda de libertad de expresión.

En el tema que nos concierne también han existido discrepancias en los medios de comunicación. Aunque la prensa general y los medios audiovisuales han dado un tinte sensacionalista a las informaciones sobre los virus, parte de la prensa especializada seria ha calibrado en su justa medida la información.

Dewdney justificaba su transparencia informativa asegurando que la mejor definición de virus era insuficiente para que personas sin los conocimientos necesarios pudieran crear monstruos informáticos. Esta afirmación presenta una pequeña fisura. Si se publica el listado completo de un virus, se corre el peligro de que programadores que hasta el momento no habían prestado atención a estos programas empiecen a experimentar con ellos. Con un listado desensamblado de un programa contaminante resulta sencillo modificar su acción destructiva. Muchos programadores que no serían capaces de crear por sí mismos un programa nocivo, por no conocer los mecanismos de manejo del *hardware* a bajo nivel, sí podrían modificar ciertas partes a su gusto. O quizá podrían modificar el modo de activación del virus, con lo que aparecerían nuevas versiones del programa, y de hecho aparecen, que complicarían su detección y desactivación. Un ejemplo claro lo ha constituido el virus Viernes-13, del que hasta la fecha circulan más de diez versiones que varían únicamente en su forma de activación.

Abogamos en esta guía por una información técnica, especializada y seria para mantener a los usuarios al tan-

to del desarrollo de estos programas, de los síntomas para detectarlos y de los medios para combatirlos. Por esta razón se ha creído conveniente no incluir listados completos de virus, sino partes de su código para que de su análisis puedan extraerse conclusiones constructivas en la lucha contra los virus.

Durante la búsqueda de documentación para este trabajo han surgido anécdotas que reflejan el impacto producido por la desinformación existente.

Un directivo de una empresa dedicada a las redes de área local, afectada por el virus Viernes-13, explicaba el método utilizado para erradicar el mal de los *diskettes* de la empresa. Pasaba el *diskette* por el campo magnético de un electroimán que había extraído de un electrodoméstico casero. Ignoramos si la información que albergaba el soporte resultaba de interés, pero está claro que desaparecería en la operación. Evidentemente, el remedio era bastante peor que la enfermedad.

Según las informaciones emitidas en televisión sobre este virus, se podía malinterpretar que se contagiaba por contacto directo del soporte magnético con el ordenador. Concretamente, la frase utilizada fue "si usted tiene un *diskette* contaminado con el virus Viernes-13 como éste (el comentarista enseñaba un *diskette* en su mano derecha) y lo introduce en su ordenador (introducía el disco en la unidad de disco), lo habrá contaminado".

Una semana más tarde recibíamos la llamada de un conocido que trabaja en una empresa de renombre en el mundo de la informática, comentándonos que había sorprendido a la señora de la limpieza realizando sus labores habituales en los ordenadores de la oficina a una distancia prudencial y con especial cuidado. Cuando éste le pidió una explicación sobre su extraño comportamiento, la empleada respondió que no quería contraer el virus.

Un periódico de renombre nacional "laureaba" las habilidades de una brillante empresa española que había desarrollado el método para erradicar el "virus de la pelotita". El fastuoso método consistía en transcribir de nuevo el sistema operativo al *diskette* contaminado con la orden SYS del DOS (*Disk Operating System*), el sistema operativo más común de los ordenadores personales. Este virus, como se detalla más adelante, coloca parte de su código en el sector de arranque del disco y el resto lo camufla en

sectores que marca como defectuosos. El eficaz método descrito en el periódico no era más que un comando del sistema que destruía la porción de código situada en el sector de arranque.

Estos increíbles casos son un exponente claro de la información que se ha recibido en nuestro país.

El alcance de este tipo de delito informático exige que las autoridades judiciales empiecen a modificar las leyes para castigar duramente este tipo de actos. Parece que el fenómeno es una fiebre pasajera únicamente avivada por el sensacionalismo de la prensa en los momentos críticos de la posible activación de los virus. Probablemente cesará en su intensidad cuando se procese a los responsables.

Si se quiere buscar un aspecto positivo a la proliferación de los virus, se encuentra en los esfuerzos que la informática mundial está realizando en el campo de la protección de los sistemas. Los planes informáticos de las empresas empiezan a contemplar medidas para evitar la acción de estos programas. Las casas diseñadoras de sistemas operativos están sacando experiencias positivas de los ataques a sus productos e intentan proteger mejor sus nuevas versiones.

2

Tipos de virus

2.1. Otros programas infecciosos

Antes de establecer la clasificación de los virus conviene identificar otros programas que generalmente se confunden con ellos. Estos programas son el *gusano*, el *caballo de Troya* y la *bomba lógica*.

Un "gusano" es un programa que se desplaza por la memoria interna del ordenador con identidad propia, a diferencia del virus, que generalmente se adhiere a otros programas. Está diseñado para que busque zonas de memoria desocupadas, donde realiza copias sucesivas de sí mismo, hasta que consigue un desbordamiento físico de la memoria. Otra diferencia importante frente al virus es que los segmentos recurrentes del programa que se generan en la memoria mantienen "comunicación" con el segmento de programa por el que fueron creados. Estos programas proliferan frecuentemente en redes públicas de comunicaciones y en redes de área local, y utilizan el correo electrónico como medio de propagación entre ordenadores de una misma red.

Erróneamente, se ha definido el "caballo de Troya" como una sección de código camuflado en el seno de un programa legítimo, que hace que éste realice servicios que están fuera de sus especificaciones. Si se repasa el origen literario del nombre de estos programas, tomado de *La Ilíada* de Homero, se comprobará que el caballo de Troya escondía en su interior a los soldados griegos que asaltaron la fortaleza de la ciudad de Troya. De esta observación se deduce el error cometido en la definición.

Así pues, denominaremos caballo de Troya al programa legítimo que porta en su seno el código pernicioso, y no al propio código, como se había hecho hasta ahora.

Como característica esencial de estos programas indicaremos que carecen del factor de autorréplica. Por ello, su código maligno sólo se activa una vez al ejecutar el caballo de Troya que lo porta. Se utiliza para extorsionar operaciones rutinarias, como el redondeo de cuentas en los procesos de actualización de cuentas bancarias. También se usa para atacar los sistemas de empresas introduciendo el programa malicioso por la puerta principal, disfrazado de publicidad o de información de utilidad. Recientemente se ha dado un caso en nuestro país. Distintas empresas españolas han recibido un *diskette* procedente de Inglaterra con información sobre el SIDA y portador de un caballo de Troya, cuyo código maligno destruía la información del disco duro.

Por último, se denomina "bomba lógica" o "bomba de tiempo" a un programa que se ejecuta al producirse un hecho predeterminado. La condición o hecho que motiva la activación es variable y comprende desde una fecha determinada, por ejemplo un viernes 13, hasta secuencias especiales de teclas. Si no se produce el suceso programado, el programa permanece oculto al usuario, sin ejercer más acción que ocupar una porción de la memoria. Cuando llega el momento determinado, ejerce la acción para la que se programó.

El motivo que ha originado la confusión de los virus con los programas camuflados en caballos de Troya o con bombas lógicas, es que muchos virus utilizan características similares a las de estos programas. Se trata, por ejemplo, de la posibilidad de estar contenidos en otros programas y permanecer ocultos a los ojos del usuario, o bien de sus condiciones de activación. Como ejemplo, cabe reseñar el virus Viernes-13. Este virus, como se detalla en el apéndice B, precisa para ejercer su acción destructiva que la fecha del sistema coincida con el día trece del mes y con un viernes como día de la semana. Por otro lado, se adhiere a la estructura de los ficheros ejecutables para ocultarse, convirtiéndolos a su vez en caballos de Troya portadores del código del virus.

En cuanto a los gusanos, la diferencia más significativa frente a los virus estriba en que los primeros ejercen su

acción de copia sucesiva en la memoria interna del ordenador, mientras que los virus realizan su función de autocopia en el soporte magnético. Esta función de copia es la que ha motivado la confusión.

2.2. Tipos de virus

En el primer capítulo se desechó una primera clasificación de los virus en benignos y malignos. En la clasificación que se ofrece a continuación, la división de los tipos se hace en función de la parte modificada por el virus en su ataque. Para ello, conviene refrescar algunos conceptos.

Una característica esencial en la propagación de los virus la constituye su mecanismo de arranque. Un virus únicamente puede llevar a cabo su acción si su código tiene oportunidad de ejecutarse. Así pues, estos programas consiguen propagarse a través de su ejecución parasitaria, modificando programas de uso frecuente a los que añade la tarea adicional de la ejecución del propio virus.

En los ordenadores personales, o PC, además de los programas de aplicación, el objetivo principal de los virus son aquellas partes del sistema operativo utilizadas con mayor frecuencia. Son las siguientes:

- El sector de arranque de la partición del disco duro. Generalmente, el sistema operativo DOS ofrece la oportunidad de partir o dividir el disco duro en particiones que alberguen otros sistemas operativos. El sector que determina la partición ocupada por el DOS es el punto habitual donde se aloja el virus.
- El sector de arranque de los discos duros o *diskettes*. El lector encontrará más información en el apéndice A.
- Los ficheros ejecutables con extensión .EXE y .COM, entre ellos el programa de sistema operativo COMMAND.COM, o intérprete de comandos del DOS.
- Los ficheros con extensión .OVL, correspondientes a las expansiones de los ficheros .EXE. El término OVL corresponde a la abreviatura de la palabra inglesa *overlay*.

Estos puntos de ataque configuran una base aceptable para la clasificación de los tipos de virus. Otro aspecto fundamental en la división de los programas contaminantes es el mecanismo de arranque utilizado en los ordenadores personales bajo DOS. El lector podrá obtener más información en el apéndice A.

Atendiendo, pues, a las partes que modifican los virus en su ataque, éstos pueden clasificarse en dos grupos:

- Virus del sector de arranque.
- Virus de programas.

2.2.1. Virus del sector de arranque

Entre los virus del sector de arranque se incluyen aquellos que son capaces de modificar el sector de *bootstrap* o arranque, ya sea el de la partición del DOS o el del disco duro o *diskette*. Como norma general, sustituyen el contenido original del sector por una versión propia para arrancar el sistema. El contenido original del sector de arranque se almacena en cualquier sector libre del disco.

Cuando se inicia una sesión de trabajo con el sistema, se ejecutará primero la versión modificada del sector de arranque y, si fuera necesario, se ejecutaría la versión original. Por este motivo, para camuflar la presencia del virus cuando se trata de leer este sector con alguna utilidad de mantenimiento de discos, el virus no destruye su contenido original.

En el sector de arranque se almacena la porción de código del virus que permanece residente en la memoria interna. Este código será el encargado de llevar a la memoria el resto del virus situado en un grupo de sectores del disco/*diskette*, que figuran como sectores en mal estado, y que realmente contiene el cuerpo del programa contaminante.

Un virus de este tipo ejerce también funciones de monitor o coordinador e interfiere la acción del sistema operativo desde el momento en que se almacena en la memoria.

Como resumen, comentaremos que el mecanismo de un virus del sector de arranque utiliza tres componentes distintos para acomodarse en el ordenador:

- El propio sector de arranque, que es reemplazado por la versión contaminada; es la puerta por donde el virus accede a la memoria.
- Un sector libre del disco donde se deposita la versión original del sector de arranque que se ha sustituido.
- Cierto número de sectores libres para depositar el cuerpo del código del virus. Tras su grabación se marcarán como sectores en mal estado.

Los ejemplos más claros de este tipo de virus son el Brain (virus del sector de arranque únicamente de *diskettes*) y el Italian o virus de la pelotita (virus del sector de arranque de discos duros y *diskettes*, y del sector de partición de discos duros).

2.2.2. Virus de programas

Dentro de esta división se incluyen los virus que son capaces de modificar la estructura de los ficheros ejecutables con las extensiones .COM, .EXE o .OVL.

Estos virus pueden insertarse al principio o al final del fichero que contaminan, dejando generalmente intacto el cuerpo del programa que los alberga.

Cuando se adhieren al final del fichero, modifican la instrucción inicial de salto del programa aunque respetan su función. Sólo en raras ocasiones inutilizan el programa portador sobreescribiendo su código ejecutable con basura.

Cuando se ejecuta un programa contaminado, el virus toma el control y se instala residente en la memoria por medio de un servicio de interrupciones del DOS. A continuación pasa el control al programa que lo porta, permitiéndole una ejecución normal. Una vez finalizada su ejecución, si se intenta ejecutar otro programa no contaminado, el virus ejercerá su función de autocopia insertándose en el nuevo programa que se ejecuta.

En general, este tipo de virus se propaga infectando un programa, que en la mayoría de los casos no está infectado, cuando se le llama para su ejecución. El virus aprovecha su posición de residente en memoria para comprobar si el programa que se va ejecutar está contaminado previa-

mente o no. Si no lo está, el virus se le adhiere; si lo está, el virus no hace nada. Este último comportamiento tiene una excepción en ciertas versiones del virus Viernes-13, que se supone un fallo de programación, como podrá comprobarse en el apéndice B.

En algún caso particular, el virus no se instala residente en memoria, extendiendo su infección con la búsqueda del primer programa no infectado del disco. Una vez encontrado, lo infecta y deja libre la memoria.

3

Casos famosos de virus

3.1. El virus de Israel o Viernes-13

Es el caso más conocido dentro y fuera del mundo de la informática, debido a su gran difusión y al protagonismo que adquiere en los medios informativos cada vez que se acerca un viernes y trece. Su gran expansión se debe a que respeta las funciones habituales de los programas que lo albergan.

La primera versión de este virus fue descubierta en diciembre de 1987, en los ordenadores de la Universidad Hebrea de Jerusalén, por lo que este virus se denomina también virus de Jerusalén, virus de Israel o virus de la Universidad Hebrea, además de Viernes-13.

Su descubrimiento se debió a lo que se supone un fallo en el diseño del programa. El virus no detectaba programas ya contaminados con extensión .EXE y, por tanto, volvía a infectarlos, llegando a alcanzar éstos un tamaño imposible de manejar por el sistema operativo DOS. El resto de los programas ejecutables sólo quedaban infectados una vez. Si un programa contaminado se ejecutaba, el virus pasaba a la memoria del ordenador, memoria de trabajo (RAM), y a partir de ese momento se contaminaba cualquier programa que se ejecutase. Si se trataba de un programa con la extensión .EXE, se contaminaba tantas veces como se utilizase. Cada infección aumentaba la longitud del programa en 2 Kb aproximadamente, encontrándose después de cierto tiempo programas que se habían contaminado numerosas veces.

El virus, totalmente desconocido, se extendió por Israel

rápidamente, debido principalmente a que este país está altamente informatizado en redes.

Los ordenadores personales mostraban claros síntomas de mal funcionamiento, manifestaban lentitud y largo tiempo de respuesta. Debido a su tamaño, algunos programas no podían ejecutarse por falta de espacio suficiente en la memoria de trabajo. Estos síntomas llevaron a los expertos pertenecientes a la Universidad Hebrea a investigar el fenómeno, hasta que a finales de diciembre de 1987, dieron con el virus. Pudieron así desactivar la pequeña bomba de relojería cuya detonación estaba preparada para el 13 de mayo de 1988, con el objetivo de borrar programas militares y científicos, e innumerables programas pertenecientes a los usuarios de ordenadores personales. La vacuna programada por los expertos de la Universidad Hebrea amortiguó su efecto.

Existen dos teorías sobre el origen y el objetivo principal del virus. Ambas hacen referencia a su fecha de activación.

La primera de ellas, y más convincente, se deduce de las instrucciones relativas a la obtención de la fecha del ordenador para su comparación. Dicha versión ignora todos los posibles viernes y trece que pudieran existir en 1987, año en que se dedicaría únicamente a la multiplicación y propagación. Esta teoría, atribuida a un origen político, juega con la posibilidad de que el virus fuese un nuevo tipo de arma lógica creada contra el pueblo judío, posiblemente por seguidores palestinos. Recordemos que el primer viernes y trece del año 1988 fue en el mes de mayo y coincidió con el cuadragésimo aniversario del final de la guerra del Yom Kippur. Las consecuencias de dicha guerra fueron la desaparición de Palestina y la constitución del Estado de Israel el 14 de mayo. Por tanto, el 13 de mayo de 1988 se celebraba el cuadragésimo aniversario del último día de la existencia de Palestina.

La segunda de las teorías, menos difundida, asegura que las especulaciones de la anterior son pura coincidencia. Basa la existencia del Viernes-13, tanto en Israel como en Estados Unidos, en que son países con buenas redes de telecomunicaciones y no por un objetivo político. Se ampara en que tal fecha es símbolo de mala suerte para la cultura anglosajona, como lo es en España el martes y trece.

La razón de que el virus no se activase durante el año 1987 se debe a una etapa de lo que se pudiera denominar incubación. Si el virus hubiera actuado en el mismo momento en que infectó un programa, su labor destructora hubiera sido mínima y al detectarse tan rápido se podría haber descubierto a su creador. Por eso, su programador alargó el período de incubación durante todo un año esperando que su objetivo abarcara el mayor campo posible.

En España se dio a conocer de forma pública en el mes de abril de 1989, al ser introducido de forma masiva e involuntaria por una revista en *diskettes* de tirada nacional. La revista no comprobó los *diskettes* antes de lanzarlos al mercado. De esta forma se infectaron todos los programas al utilizar el *diskette* y sólo algunos usuarios advirtieron un comportamiento anormal en su ordenador que les hizo sospechar. La editorial retiró los ejemplares que quedaban por vender, pero el virus ya estaba en la calle. Con ayuda de las copias "piratas" existentes, su expansión fue rápida. La dirección de la revista denunció el caso ante los juzgados como sabotaje.

La alarma había saltado y los medios de comunicación acrecentaron el temor, que se extendió más rápido que el propio virus. La misma revista, en el número siguiente, y otras empresas informáticas ofrecieron distintas vacunas y antídotos de forma gratuita, así como direcciones, teléfonos y BBS con los que ponerse en contacto para ampliar la información sobre el virus. En los medios de comunicación audiovisuales se podían encontrar soluciones tan tajantes como no encender el ordenador en todo el día en la fecha fatídica, o más coherentes, como cambiar la fecha.

El virus se había extendido por todo el mundo, pero no era el mismo. Por un lado, los sistemas operativos habían evolucionado, haciendo difícil predecir los efectos de manera detallada. Por otro, existían nuevas versiones mejoradas del Viernes-13 con efectos distintos a los del original. Los defectos que hasta entonces permitían su detección habían desaparecido.

En octubre de 1989, el Viernes-13 no se encontraba solo para actuar devastadoramente en una fecha señalada. Otro virus, llamado virus del Día de Colón (o Datacrime), acechaba los ordenadores IBM y compatibles. El virus estaba pro-

gramado para actuar el 12 de octubre, fecha del 497 aniversario del descubrimiento de América. Se ha especulado bastante sobre si su desarrollo tuvo origen en Europa y si su misión era invadir los sistemas informáticos norteamericanos. Llevaba a cabo su efecto destructor atacando la tabla de localización de ficheros en el disco (FAT) y haciendo difícil la reutilización de los datos almacenados en él.

Se trataba de dos virus importantes, por sus efectos, en la misma semana. Un riesgo que no podía permitirse un centro como la Agencia espacial norteamericana (NASA), que decidió aplazar el lanzamiento del transbordador espacial "Atlantis" con la sonda Galileo a bordo, como medida de precaución.

3.2. El gusano de la NASA y el Pentágono

Se trata del caso más espectacular de contaminación informática producido por un gusano. Su entorno fue la red ARPANET (*Advanced Research Projects Administration Network*), con miles de terminales en varios continentes y en lugares tan estratégicos como son el Pentágono o la NASA.

Esta misma red se infectó en octubre de 1980 con un virus de procedencia desconocida que la dejó fuera de servicio durante 72 horas hasta que los técnicos reestablecieron las comunicaciones. Este nuevo gusano demostró la vulnerabilidad que continuaba existiendo en los sistemas de seguridad de uno de los centros estratégicos más importantes.

Repasando cuidadosamente los detalles de este nuevo caso, se puede llegar a una conclusión alejada de la casualidad. La infección fue originada por un joven de 23 años llamado Robert Tappan Morris, estudiante de informática en la Universidad de Cornell, Ithaca (Nueva York). Era un estudiante brillante y admirado por sus compañeros. Su apariencia era la de un genio: gafas enormes, pelo descuidado hasta los hombros y gustos atípicos. Claro prototipo del estudiante de informática, encantado de quedarse las horas muertas frente al ordenador intentando resolver errores de un programa, "pirateando" la última

aplicación que caía en sus manos o indagando en las innumerables redes.

Su padre, aclarando más el caso, era un prestigioso científico del gobierno, encargado de garantizar la seguridad de la red nacional de distribución de información. Se supone que trabajaba también para el contraespionaje norteamericano. Era un alto cargo en el Centro nacional para la seguridad de los ordenadores, en Bethesda (Maryland).

Precisamente fue quien desarrolló el sistema UNIX conocido como Berkeley 4.3 que utiliza la red ARPANET. Un colaborador del padre, Eric Allman, había insertado en el programa una puerta oculta para el correo electrónico. Este tipo de correo envía cartas pasivas, es decir, que aparecen en la pantalla como cualquier otro texto. De esta forma quedaba un pasillo libre de controles por donde podían circular los programas.

La red ARPANET es una de las redes más grandes del mundo. Está conectada a su vez con otras no menos importantes, basadas en máquinas SUN y VAX con sistema operativo UNIX V, Milnet e Internet, que también sufrieron la infección. Fue fundada por la Agencia de proyectos de investigación avanzada del Ministerio de Defensa, para tratar material no clasificado entre universidades y centros de investigación públicos y privados de Estados Unidos y muchos otros países.

Los hechos pudieron ocurrir de la siguiente forma. Tras meses de estudio y preparación, en la primera semana de noviembre de 1988 Robert T. Morris decidió hacer la prueba final. Con la intención de ocultar un programa en la red a la que pertenecía su universidad, Robert se puso manos a la obra la noche de un miércoles. Su programa constaba de 50000 bytes que correspondían a 5000 líneas de código, un programa bastante largo.

Cuando Robert terminó de cenar y volvió a sentarse frente a su terminal con la intención de averiguar lo que estaba pasando con su programa, descubrió que la prueba se había desbordado. Las miles de líneas de programa actuaban activando el correo electrónico, como había supuesto Robert T. Morris. Se copiaba en la memoria del ordenador y se "autoenviaba" a todos los terminales que aparecían en la lista de correo del mismo. Lo que esperaba es que esta operación, al repetirse en cada ordenador,

se volviese a enviar y copiar incluso en aquellos ordenadores por los que ya había pasado. En pocas horas el programa viajó, ida y vuelta, por los mismos ordenadores miles de veces, copiándose una vez más en cada ordenador. Esto supuso una saturación de las líneas de comunicación y de las memorias de los ordenadores conectados a la red, que quedaron bloqueados.

Cuando Robert T. Morris reaccionó era demasiado tarde. Llamó a un amigo para que diese la voz de alarma a la red, mientras él intentaba solucionarlo desde su terminal con sus conocimientos sobre el programa y lo que estaba sucediendo.

Más de 6000 ordenadores quedaron infectados. Entre ellos, los del Pentágono, la NASA, el Mando aéreo estratégico (SAC), la Agencia nacional de seguridad (NSA), el Ministerio de Defensa, los laboratorios Lawrence Livermore de Berkeley (California), donde se desarrollaban varios componentes de la Iniciativa de Defensa Estratégica, también llamada "guerra de las galaxias", y las universidades de Princenton, Yale, Columbia, Harvard, Illinois, Purdue, Wisconsin y el Instituto de Tecnología de Massachussetts. Incluso se llegó a infectar ordenadores de la República Federal de Alemania y Australia.

Pudo ser mucho peor. Los técnicos subsanaron el problema con relativa facilidad, aunque el programa puede continuar hibernando en alguno de los buzones electrónicas a los que fue enviado. El Pentágono ordenó inmediatamente un estudio para determinar las medidas de seguridad que había que añadir a las ya existentes, y una valoración de las pérdidas, no sólo económicas.

Como ocurre en este tipo de delitos, es difícil descubrir al culpable, a no ser que él mismo se delate. La legislación existente para estos delitos está muy verde, pero no por ello Robert T. Morris quedó exento de culpa. Se le ha tomado como "cabeza de turco" para impedir que se realicen posteriores prácticas similares. Fue acusado de violar el código de Integridad Académica de la Universidad de Cornell, y está siendo juzgado en la actualidad. En una investigación realizada por su propia Universidad junto con la policía del Estado, se le encontraron más de 430 códigos secretos para entrar en otros tantos ordenadores de instituciones y universidades, como las de Cornell y Stanford, y otros ordenadores personales unidos a la red.

3.3. Casos paralelos de gusano

En Suecia se consiguió paralizar un gusano a tiempo. Su creador era también un estudiante, de la Escuela Técnica Superior de Lund, perteneciente a la ciudad universitaria que se encuentra al sur de Suecia. Estos son los únicos datos que se conocen del delincuente.

El intento fracasó por un fallo de ortografía. El saboteador se había hecho con una copia del programa de Robert Tappan Morris y la modificó. Consiguió entrar en la red a través del ordenador de la Universidad de Linkoeping, que está conectada a los sistemas informáticos de Suecia, Noruega y Finlandia. La falta de ortografía alertó casualmente a los responsables de esta universidad, situada a unos 180 Km al sudoeste de Estocolmo. Se ordenó una investigación y de esta forma se pudo salvar el sistema.

Existe un caso anterior al gusano de la NASA, y de efectos parecidos, que ocurrió en diciembre de 1987. Unos estudiantes pertenecientes a la universidad alemana de Klausthal quisieron felicitar la Navidad a todos los miembros de la cadena europea EARN (Red europea académica y de investigación), creada en 1984 con la participación de varios países europeos, entre los que se incluye España.

No se trataba exactamente de un virus, era tan sólo el envío de un mensaje a través de un programa a todos los terminales conectados a la red. Como en el famoso caso de Robert Tappan Morris, los estudiantes perdieron el control del programa. Con un mecanismo equivalente al del "círculo o pirámide de oro" o las cadenas de cartas, el mensaje se fue enviando a todos los terminales pertenecientes a la red a gran velocidad, creándose un bucle sin fin que saturó la red y bloqueó todos los ordenadores conectados.

Japón no ha quedado al margen. En septiembre de 1988 la principal red japonesa de ordenadores conectada a nivel nacional por la empresa NEC Corporation, resultó infectada con un gusano que alteraba el funcionamiento de los ordenadores y la información que en ellos se almacenaba. NEC tiene un sistema de interconexión de ordenadores personales conocido como PC-VAN, con 47000 afiliados que intercambian información o mantienen correo electrónico. Precisamente el correo fue la cau-

sa, como en los casos anteriores, de la infección de un buen número de ordenadores.

Estos hechos obligaron a que las autoridades se tomasen en serio la búsqueda de un programa-vacuna. La preocupación llevó al Ministerio de comercio internacional e industria (MITI) a destinar una partida especial de 750000 dólares en sus presupuestos del año 1989, para detectar los programas que habían originado los virus y sus posibles remedios. Pero no evitó que entre octubre y noviembre de 1989 cuatro universidades japonesas fuesen invadidas por gusanos atribuidos a grupos antinucleares. Invadieron los ordenadores del Laboratorio de Física de alta energía de la prestigiosa Universidad de Tsukuba, así como los departamentos de Física de las universidades de Tokio y Osaka y del Instituto de Estudios Nucleares. Todos los ordenadores descritos estaban conectados a la red Hep-Net, para intercambio internacional de informaciones de física de alta energía. Los daños no fueron de gran importancia, ya que el objetivo era únicamente propagar un mensaje que apareciese cientos de veces en todos los ordenadores: "Gusanos contra asesinos nucleares".

Los virus y gusanos se han convertido en el arma de protesta de grupos ecologistas. En las pantallas de los terminales de ordenadores de ciertos centros aparecen mensajes contra la labor que allí se realiza.

Otro ejemplo de este tipo ocurrió en octubre de 1989 en Estados Unidos con el lanzamiento de la sonda espacial Galileo con rumbo a Júpiter. Antes del despegue, efectuado con el transbordador espacial Atlantis desde Cabo Cañaveral, en Florida, la NASA confirmó la infección de una de sus redes por un gusano que iba diseminando mensajes antinucleares relativos al Galileo.

El gusano no tenía como objetivo acción destructiva alguna, pero el temor radicaba en que algunos datos reales fueran sustituidos por "basura". Al menos tres centros, el Centro de vuelo espacial de Goddard, en Maryland, el Laboratorio de propulsión a reacción de Pasadena y el Centro de investigación Ames, cerca de San Francisco, llegaron a infectarse por el programa introducido en Internet.

La protesta ecológica se refería al riesgo de una explosión del Atlantis, que provocaría una radiación debida al reactor de plutonio que impulsaba al Galileo. Aunque

fueron grupos activistas americanos quienes anunciaron antes del lanzamiento su intención de sabotear el despegue, siguiendo la pista dejada por el gusano en la red Internet se dedujo que el origen de la infección provino de Europa, y más concretamente, de Francia o Suecia.

3.4. El virus Brain

Este es un caso digno de reseñar por tratarse de uno de los virus más extendidos entre los usuarios del sistema operativo DOS y porque finalmente se pudo identificar a sus autores, los hermanos Basit y Alvi Amjad, de Lahore, Pakistán.

La primera versión del virus se instalaba en el sector de arranque y consistía en varios sectores marcados como en mal estado. Aparentemente no producía daños. Cambiaba la etiqueta de volumen de los *diskettes* de 5,25 pulgadas, que contenían sistema operativo, por la de "(c) Brain". No infectaba el disco duro y solamente atacaba los *diskettes* con sistema operativo de versión inferior a la 2.0, destruyendo pequeñas cantidades de datos, sólo si los discos estaban casi o totalmente llenos. Pero como ocurre con la mayoría de los virus, empezó a ser molesto y aparecieron nuevas versiones mejoradas que inutilizaban datos almacenados e infectaban el disco duro y todas las nuevas versiones del sistema operativo.

Aunque se estipula que se creó en 1986, se dio a conocer el 16 de mayo de 1988 en Estados Unidos cuando un periodista del *Journal-Bulletin* de Providence, Rhode Island, no podía recuperar un fichero almacenado en el *diskette* en el que había guardado el trabajo de varios meses. Llevó entonces el *diskette* deteriorado al departamento de sistemas de la casa que lo fabricaba, donde un analista detectó que el bloque de inicialización del disco contenía un programa vírico. Uno de los ingenieros de sistemas de Providence Journal Corporation, Peter Scheildler, llegó a provocar involuntariamente varias reinfecciones al intentar combatir la infección en los *diskettes* de la compañía, de las oficinas de agencias estatales y de los ordenadores personales de los empleados.

El virus, actualmente en activo, se caracteriza por un mensaje que aparece en el primer sector del *diskette* con-

taminado. El mensaje, que varía según la versión del virus, es similar al siguiente: "Welcome to the Dungeon ... (c) 1986 Brain & Amjads (pvt) Ltd ... VIRUS_SHOE RECORD V9.0 ... Dedicated to the dynamic memories of millions of virus who are no longer with us today - Thanks GOODNESS!! ... BEWARE OF THE er... VIRUS...". La traducción podría ser: "Bienvenido a la mazmorra ... [Marca del *copyright* de los hermanos Amjad], [posible fecha de creación del virus] 1986 [versión del programa]... Dedicado a las memorias dinámicas de los millones de virus que ya no están con nosotros [se supone que por haber sido detectados y desactivados] - ¡GRACIAS A DIOS! ... CUIDADO CON EL... VIRUS...". Esta última frase debe entenderse como el último suspiro, antes de la muerte de un actor en una representación dramática, que pretende avisar a alguien de un peligro, resultado, sin duda, del buen humor de los creadores del programa.

En algunas versiones, el mensaje menciona un número de teléfono de una compañía de ordenadores pakistaní. El ingeniero de Providence Journal Corporation se puso en contacto con dicho teléfono, que correspondía a la empresa de los hermanos Amjad, quienes, tras excusarse de los daños ocasionados, afirmaron que el virus se escribió originalmente para que les ayudara a seguir el rastro de las copias "pirateadas" del *software* cuyo *copyright* disponían desde 1986. También aseguraron que no comprendían cómo se había extendido de esa forma el virus, alejado de las copias de sus programas, ni cómo había llegado hasta Europa y Estados Unidos, ya que sólo debía afectar a aquellos usuarios que utilizasen alguno de sus programas de forma pirata.

3.5. El *Chaos Computer Club* (C.C.C.)

Existen ciertos foros de reunión de los personajes que se dedican a la intromisión en redes de comunicación y a otro tipo de fechorías. El más famoso tiene su sede en Hamburgo, R.F.A., y es conocido por sus tres siglas, C.C.C., que corresponden a *Chaos Computer Club* (club del caos informático).

Además de organizar conferencias sobre las acciones que llevan a cabo, explicando detalles de la metodología

seguida para perpetrar sus delitos, esta asociación presta incluso asistencia jurídica a sus miembros. La mayor parte de las veces sus miembros resultan absueltos, ya que no puede acusárseles de infringir una ley que en la mayoría de los casos no existe y sobre la que no hay sentada jurisprudencia.

Otra de sus características es la divulgación de sus logros, y el anuncio de sus próximas objetivos. Llegan a retar a compañías informáticas para que prueben sus sistemas de seguridad frente a sus técnicas para eludirlos.

Aunque con sus actos ofrecen otra imagen muy distinta, lo cierto es que muestran desinterés por los datos almacenados y afirman que su único objetivo es demostrar la extrema vulnerabilidad de los sistemas de seguridad en las redes de bancos de datos. Esta argumentación parece tener veracidad ya que, después de acceder a una red, ponen a disposición de los centros afectados un informe sobre las entradas ilegales a su sistema, con el que, dicen, se ayuda a eliminar sus deficiencias y a incrementar su seguridad.

Uno de sus logros más sonados fue la intrusión en la red de análisis de física espacial, SPAN, logrando una espectacular intromisión en los sistemas secretos de ordenadores de la agencia espacial norteamericana, NASA, y de numerosos institutos de investigación espacial, biomolecular y nuclear. Algunos de los centros afectados fueron el Max Planck de física nuclear, el centro de investigación de Fermilab de Batavia, en las cercanías de Chicago, y el Laboratorio Europeo de Biología Molecular. Cerca de cincuenta sistemas entre Europa y Estados Unidos resultaron "asaltados".

Según los responsables del C.C.C., la puerta de acceso en el *software* de la red se debía a un error en la programación del sistema de control de acceso. Por esta puerta, los componentes del club fueron introduciendo paulatinamente programas propios para tener acceso continuado a los bancos de datos. Estos programas, codificados de forma que no pudieran ser descubiertos, les permitieron disponer de datos secretos pertenecientes a los institutos en la vanguardia de la investigación mundial en los citados sectores.

La agencia espacial norteamericana señaló en un comunicado que la red tenía el propósito de facilitar, a "indivi-

duos apropiados", el acceso a datos no secretos de la NASA. De forma que cualquier persona u organización que realizase investigaciones relacionadas con la NASA podía tener acceso a ella, pero si este acceso lo realizaban personas no autorizadas, se corría el peligro de alterar los datos.

El C.C.C. negó que durante la espectacular intromisión se modificasen datos y agregaron que una modificación de ese tipo iba contra la ética del club y del objetivo por el que fue creado.

3.6. Casos en España

En España no se han producido casos tan espectaculares, si bien las empresas no están muy dispuestas a desvelar este tipo de accidentes, a no ser que el suceso salga a la luz. Un ejemplo claro es el de la revista que con su *diskette* colocó el virus Viernes-13 en su tirada de forma involuntaria. El caso ya ha sido comentado al comienzo de este capítulo.

Una de las instituciones que se ha visto afectada por un virus de los considerados "benignos" es la Bolsa de Barcelona. Este virus residía en uno de los discos duros de un ordenador de su departamento de comunicaciones. Se trata del mismo virus que había infectado meses antes los ordenadores de la Universidad Politécnica de Barcelona, y cuyo nombre más popular es "virus de la pelotita". Su acción consiste en alterar la presentación por pantalla de cualquier programa que se esté ejecutando, con un carácter que rebota constantemente en las cuatro paredes del terminal de vídeo. Su "benignidad" desaparece con el retardo del tiempo de respuesta del ordenador que supone un virus residente en memoria. En pocos días proliferaron las vacunas y la epidemia desapareció.

Pero si tenemos que destacar un caso por las repercusiones que pudo llevar consigo, éste es la alerta que se produjo el día anterior a la celebración de las elecciones generales del 29 de octubre de 1989. A principios de esa semana se recibió un aviso anónimo en el Ministerio del Interior, comunicando que alguien podía intentar boicotear la red que iba a ocuparse de gestionar los datos de las elecciones. Esta red incluye un ordenador central y varios

cientos de terminales, entre los que hay 140 ordenadores personales, que suponen un montón de puntos de acceso para un saboteador. Los técnicos de la empresa encargada de la recogida y recuento electrónico de los votos emitidos, tuvieron que elaborar a toda prisa un sistema de seguridad. El único motivo de alarma fue la detección de una serie de signos ilógicos en los campos de fecha de los ordenadores.

El complejo informático que se utilizó en las últimas elecciones europeas, denominado Tritón, también fue sometido a un cuidadoso proceso de "lavado" para evitar cualquier tipo de anomalía. Incluso se realizó una investigación sobre todas aquellas personas que pudieran tener acceso a puntos sensibles de la red.

El último de los casos que ha llamado la atención dentro de nuestras fronteras lo ha causado un caballo de Troya a mediados del mes de diciembre de 1989.

El código contaminante se encontraba camuflado en un programa de un *diskette* que contenía diversa información sobre el SIDA. El "regalo" fue enviado en un sobre blanco y sin remite a diversos técnicos y operarios de grandes empresas españolas, ministerios, hospitales y departamentos del Estado, cuyos nombres figuraban como suscriptores de revistas para usuarios del IBM PC y compatibles. Se trata del mismo camino que llevan todos aquellos *diskettes* que se reciben en estas empresas como promoción publicitaria y que no siempre siguen las medidas preventivas de seguridad que se aconsejan.

Por el franqueo se sabe que dicho sobre procedía de Gran Bretaña, donde se supone que hay distribuidos más de 20 000 *diskettes* llamados Aids (SIDA). La policía inglesa investiga las pistas existentes sobre estos delincuentes internacionales.

El *diskette* contiene una serie de instrucciones en inglés para activar el programa y seis puntos breves donde se explican sus objetivos. Entre ellos se indica que está diseñado para usarse en microordenadores IBM, PC, XT y compatibles con disco duro, con versión del MS-DOS igual o superior a la 2.0 y con un mínimo de 256 Kb de RAM.

El programa toma aleatoriamente un número entero que, al coincidir con el número de veces que se ha encendido el ordenador después de haber instalado el progra-

ma, activa el virus. Existen varias hipótesis sobre cuál es el funcionamiento real de este programa, pero todas concurren en que se trata de un caso claro de chantaje. El programa crea un fichero con nombre Cyborg Doc. Cuando se accede a éste, aparece en pantalla un mensaje que exige al usuario el envío de una cierta cantidad de dólares a un apartado de correos de Panamá. El mismo mensaje aconseja no desconectar el ordenador porque sería destruida toda la información contenida en el disco duro.

El código oculto en el caballo de Troya impide que éste se reinstale más de una vez sobre el mismo ordenador y anima al usuario a que realice copias y las distribuya entre sus amistades. Ya se han desarrollado antídotos contra este programa, y se ha detectado que no produce el efecto indicado por sus autores cuando se apaga el ordenador.

4

Medios de contagio y formas de evitarlo

4.1. Medios de contagio

Antes de estudiar las medidas para detectar y protegerse de este tipo de programas, es conveniente que se recuerden los medios de contagio más frecuentes elegidos por los virus para ejercer su acción de expansión.

El medio más susceptible de contagio lo constituyen las copias ilegales de *software* que circulan entre los usuarios. Ya se habló en capítulos anteriores acerca del precio del *software* y la consecuencia de copias ilegales que se deriva de este factor. Entre los pequeños usuarios de ordenadores personales está muy extendida la copia "pirata" para conseguir todo tipo de programas. Este detalle no se le escapa a los fabricantes de virus, que aprovechan la masiva circulación de *diskettes* para colocar sus programas contaminantes.

El segundo foco de contagio lo constituyen el *shareware* y el *freeware*. Recordemos que estos vocablos hacen referencia a los programas de uso común por los que, bien se paga una pequeña cantidad en concepto de manual o de derechos de autor simulados, en el primer caso, bien no se paga nada por ellos, en el segundo.

Este tipo de *software* tiene gran aceptación entre los usuarios de ordenadores personales, factor que añadido al ya comentado precio del *software* comercial, ha puesto de moda un nuevo concepto de biblioteca informática, el BBS.

Los BBS (*Boulletin Board Systems*) son unos clubs de *software* de dominio público a los que se accede por vía

telefónica utilizando una tarjeta *modem* de ordenador personal. En ellos pueden adquirirse todo tipo de programas no comerciales pertenecientes a los grupos antes mencionados. Alguno de estos clubs exige el pago de una pequeña cuota de inscripción para conseguir programas, mientras que otros solicitan al futuro socio la inclusión de nuevos programas en la biblioteca en concepto de ingreso.

La extracción de este tipo de programas por vía telefónica se realiza mediante la compactación de los ficheros que componen la aplicación en un único fichero, cuya extensión generalmente es .ARC. En este fichero se encuentran todos los ficheros de programa o de datos que componen una sola aplicación. El motivo de realizar este empaquetamiento es economizar la llamada al BBS en tiempo. Precisamente este factor de compactación hace peligroso conseguir *software* por este medio.

Los responsables y administradores de las bibliotecas realizan verdaderos esfuerzos para depurar y comprobar todos los programas que llegan a sus clubs, pero la creciente variación de los tipos de virus puede llegar a hacer inútil toda labor de desinfección.

El tercer foco de esparcimiento lo componen las redes públicas de ordenadores. Si bien no son una fuente muy común de virus de PC, sí constituyen un nido abundante de gusanos, caballos de Troya y bombas lógicas. Además, la trascendencia de las epidemias es mucho mayor al ser más abundante el número de ordenadores afectados.

Estas redes ofrecen una ilimitada gama de puertas principales de acceso, restringidas únicamente por palabras o claves de acceso. Por otro lado, sus sistemas de seguridad son bastante cuestionables, como se ha demostrado continuamente con intrusiones de todo tipo. A esto hay que añadir la vulnerabilidad de las utilidades de correo electrónico, que son también el objetivo principal para el esparcimiento de estos programas.

En los últimos meses se está desarrollando una nueva forma de contagio cuyo fin es el chantaje. Consiste en utilizar *diskettes* con vistosa presentación, que se envían por correo a empresas y organismos oficiales. Su contenido puede hacer referencia a demostraciones de nuevos programas o bien a información técnica de algún tema de actualidad. Bajo una presentación aparente se esconde un programa contaminante que en algún momento informa

al usuario de su presencia y de los posibles efectos en el caso de que no se pague una cantidad por el antídoto.

Una vez enumerados los posibles medios de contagio y antes de hacer una relación de las medidas preventivas contra el virus, debe quedar claro que no existe ninguna medida completamente efectiva para evitar el contagio de programas contaminantes. No se puede llegar a la completa prevención, pero sí reducir bastante el riesgo de contagio.

4.2. Medidas de protección

Se diferencian dos tipos de medidas de protección: las medidas preventivas propiamente dichas, que se deben seguir desde un principio cuando se adquiere un ordenador o cuando existe la plena seguridad de no estar infectado, y las medidas de contingencia, que se deben tomar en el mismo momento en que se detecta el virus, para evitar su mayor propagación.

El grupo de medidas de prevención no se debe seguir cuando se tenga duda o conocimiento de la existencia de un virus en la memoria del ordenador o en alguno de los discos, esto no haría sino propagar más su infección. En tal caso se debe pasar directamente al segundo grupo de medidas, las de contingencia, y seguirlas estrictamente en el mismo orden en el que aparecen.

Dentro del primer grupo de medidas se realiza otra subdivisión en dos tipos claramente diferenciados: las medidas que evitan que el ordenador se contagie y las que averiguan si el ordenador, o los programas que se utilizan, están infectados o no. En caso afirmativo se pasa al segundo grupo, el de medidas de contingencia.

4.3. Medidas de prevención

No temer a los virus

La primera de dichas medidas es, sin duda, la más fácil por su sencillez pero, a su vez, la más importante: no temer a los virus. Se trata de una medida psicológica aplicable al impacto y la repercusión de este tema en los usuarios inexpertos. Un virus no es más que un

programa y su parecido con sus homónimos biológicos es que no actúa ni se reproduce salvo en situaciones favorables para ello, es decir, si no se ejecuta como cualquier otro programa. Los daños que ocasiona el virus se centran en el *software* que se encuentre accesible en el momento de su ejecución.

El virus no es culpable de todas las anomalías o fallos de cualquier tipo que pueden aparecer en el ordenador, sobre todo si se trata de deficiencias de *hardware*.

Por todo ello, uno de los medios más efectivos para combatir esta plaga es la información y concienciación de la existencia de estos programas.

El elemento del *software* más importante es el sistema operativo; sin él, el ordenador pasa a ser una herramienta inútil. Este hecho no pasa desapercibido a los creadores de virus, por lo que el sistema operativo se convierte en uno de sus principales objetivos. De ahí que se deba poner·principal interés en las medidas que prevengan de estos programas que son constantemente utilizados por el ordenador.

Todas las medidas què aquí se incluyen deben realizarse con un sistema operativo no contaminado. De otro modo no sólo pueden resultar vanas, sino que estarán facilitando la labor de expansión del virus.

Utilizar siempre un sistema operativo fiable

De la misma forma que al activar un ordenador su primera acción es buscar el sistema operativo, la primera medida práctica que se debe tomar es tener la seguridad de que se está utilizando un sistema operativo que se encuentra en perfectas condiciones. Las consecuencias de un fallo por esta causa pueden ser graves, porque cualquier programa que se utilice a continuación quedará contagiado e incluso inutilizable, según sea el tipo de virus que le afecte.

Hacer una copia en *diskette* del sistema operativo

Para los ordenadores personales, se debe tener un *diskette* con un sistema operativo copia del original, que será propiedad del usuario y no deberá utilizar otra persona, ya que se perdería la seguridad plena. Siempre deberá llevar la etiqueta protectora, o la pes-

taña contra escritura en su posición de sólo lectura. Se utilizará para el arranque siempre que se siga manteniendo la confianza en él. Dicho *diskette* será el que se utilice en cualquier ordenador ajeno, de dominio público o propio, considerando propio el ordenador del que se tenga la seguridad de que no lo utilizará ningún otro usuario sin su consentimiento. Esta medida entraña las dos siguientes.

No grabar un sistema operativo en el disco duro a partir de un *diskette* dudoso

No se debe grabar un sistema operativo en el disco duro a partir de un *diskette* que no sea original o copia verificada del mismo, correspondiente al sistema.

Apagar y volver a arrancar un ordenador ajeno con el *diskette* propio

Siempre que se vaya a utilizar un ordenador personal, de los denominados anteriormente como ajenos o de dominio público, se debe apagar y volver a arrancar con el *diskette* de su propiedad. No es aconsejable aprovechar una situación en la que el ordenador ya tenga el sistema operativo en memoria.

El primer paso de la mayoría de los virus consiste en instalarse en la memoria interna del ordenador, donde se almacenan todos los programas que se ejecutan y desde donde un virus puede dirigir todas sus operaciones devastadoras y de reproducción. Este tipo de memoria sólo permanece activa el tiempo durante el que está encendido el ordenador. Por tanto, una sencilla forma de evitar que un virus siga actuando es apagar el aparato.

No trabajar con discos originales

No se debe trabajar nunca con discos originales. Cuando se va a probar un nuevo paquete de *software*, lo primero que se debe hacer es asegurarse de que se tiene un sistema operativo no contaminado en memoria. Una vez comprobado, se ha de hacer una copia de seguridad, archivando el original en un lugar seguro. Si se ha realizado esto, la pérdida de la aplicación por culpa de un virus supondría únicamente volver a realizar una copia desde el disco original archivado.

Comprobar la envoltura de los productos de *software* adquiridos

Todas las adquisiciones de *software* deben venir en su envoltura original. Si se tiene como política no aceptar copias piratas, se debe hacer hincapié en pequeños detalles como éste. Una de las características de los discos originales es su propaganda, que aparece tanto en la etiqueta diferenciadora como en su envoltura. Es fácil asignar cada *diskette* a su envoltura respectiva; por tanto, hay que desconfiar de los *diskettes* que no guarden esta relación.

El disco duro es, hoy por hoy, un elemento de *hardware* casi imprescindible por su comodidad, capacidad y velocidad de acceso. Por ello, a este elemento, así como al sistema operativo, se le debe dar un mimo especial, principalmente por la gran cantidad de datos almacenados. Las medidas siguientes hacen referencia al mantenimiento del disco duro.

No usar el disco duro como único lugar de almacenamiento

No es conveniente utilizar el disco duro como único lugar de almacenamiento. Debido a las características anteriormente mencionadas, los programas y datos que en él se almacenan son los de uso más corriente y, consecuentemente, con mayor tendencia al contagio.

Comprobar los *diskettes* antes de copiarlos en el disco duro

No se debe copiar directamente de un *diskette* al disco duro. Todos los *diskettes* deben pasar unas pruebas de seguridad dependiendo de su procedencia, original o dudosa, antes de proceder a introducir los datos en una zona tan delicada. Una posible infección en el disco duro puede resultar bastante difícil de combatir debido a la gran cantidad de información que contiene.

Proteger los discos contra escritura

Cuando no se espera realizar operaciones de escritura, es aconsejable proteger los discos contra escritura. La información que se almacena en un disco se puede dividir en tres clases: los programas escritos en código

fuente, los programas en código máquina o ejecutable (.EXE, .COM, .OVL) y los datos o ficheros de trabajo. Como ya se ha indicado en varias ocasiones, sólo se pueden ver infectados, no dañados, los programas ejecutables. Por otro lado, los programas pertenecientes a la primera clase, una vez compilados no tienen otro uso que su almacenamiento para posibles modificaciones posteriores, por lo que el grado de utilización es insignificante.

Igual que hemos diferenciado los programas ejecutables de los datos, debemos diferenciar los discos que contengan dichos programas ejecutables y no necesiten de operaciones de escritura, de los discos que contengan los ficheros de datos, que no pueden infectarse directamente y cuya lectura/escritura es casi constante. Los primeros deberán llevar siempre una etiqueta física que evite su escritura, o su pestaña de protección de escritura activada. De esta forma se impide su contagio. Los otros se caracterizan por la actualización continua de sus datos y no deben llevar protección.

De este modo, se tendrán *diskettes* que almacenan programas fuente, *diskettes* que sólo almacenan programas ejecutables y *diskettes* con los datos que utilizan dichos programas ejecutables. Estos dos últimos serán de utilización conjunta, es decir, el *diskette* que contenga un programa ejecutable, una vez cargado en memoria, necesitará el *diskette* que contenga sus ficheros de datos.

En el disco duro se presenta un problema diferente, ya que no se puede proteger contra escritura de una manera física y se debe hacer a través del *software*, lo que implica una protección algo más vulnerable. En el apéndice D se podrá encontrar un programa de protección lógica del disco duro.

Asignar al disco duro la letra de identificación E

Existe otra posibilidad de protección lógica del disco duro que consiste en asignar como letra de identificación del disco duro la E, en lugar de C, propia del disco duro en los ordenadores personales. Las letras A y B están reservadas para posibles unidades de *diskette*. Se trata de una medida poco efectiva ya que la mayoría de los virus localizan la unidad de disco duro por defecto, es decir, no dirigen su ataque contra un nom-

bre de unidad específico. Además, con esta medida se obliga a cambiar la unidad en todos aquellos programas que hacen referencia al disco duro, lo que convierte esta tarea en pesada y tediosa.

Asignar únicamente el atributo de sólo lectura a los ficheros ejecutables

Otra medida preventiva consiste en cambiar los atributos de los ficheros ejecutables dejándolos únicamente con atributo de sólo lectura. La efectividad de esta medida es similar a la de la anterior, pues se trata de una medida de *software;* es decir, de la misma forma que el usuario puede cambiar el atributo del programa, el virus puede también variarlo y una vez cambiado introducirse en él. Aunque la medida no sirva para todos los virus, sí puede desenmascarar alguno de ellos.

Colocar el fichero COMMAND.COM fuera del directorio principal y localizarlo con el comando PATH

El fichero COMMAND.COM, debido a sus características de programa ejecutable y de sistema, es un blanco perfecto. Es el intérprete de los comandos que se van a solicitar continuamente al sistema. Por esta razón es el objetivo principal de muchos virus. Su captura y manipulación, sobre todo en los comandos de entrada/salida e interrupciones, aumentan el poder expansivo del virus en el ordenador.

La medida preventiva más simple consiste en colocarlo fuera del directorio principal en algún subdirectorio, y localizarlo mediante el comando PATH del DOS en el fichero AUTOEXEC.BAT. Resulta poco efectiva ya que al ejecutarse dicho fichero por lotes, quedan definidos por defecto los subdirectorios incluidos en la orden PATH.

Cómo programa ejecutable, se le pueden aplicar todas las medidas que se han explicado para los programas de este tipo.

Realizar copias de seguridad periódicamente

La realización periódica de copias de seguridad (*backup*) es una medida bastante efectiva, al menos para la seguridad de los datos. La copia de seguridad debe ser

una rutina obligatoria incluso fuera del entorno de los virus.

Bajo la amenaza de los virus, las copias de seguridad van a desempeñar dos importantes funciones. Por una parte, una labor restauradora de los datos en caso de daños. Por otra, y como se verá más detalladamente en el apartado 4.4, una labor de detección, porque permite la comprobación entre dos copias del mismo fichero, de parámetros tales como tamaño, atributo, fecha y hora de compilación o de la última modificación, etc.

Tampoco esta medida es completamente efectiva. Una de las versiones del virus Viernes-13, concretamente la llamada Century-B, tiene la característica de esperar a que se realice un *backup* y, tomando él la dirección del comando, llenar los discos de información basura en lugar de grabar los programas que se encuentran en el disco.

Hacer un sistema rotativo de copias

No es aconsejable hacer *backup* sobre las copias de seguridad anteriores, se debe realizar un "ciclo". Lo ideal es hacer un sistema rotativo de copias, evitando con ello que la única copia de que se disponga pueda ser la realizada cuando el ordenador ya estaba contaminado.

El número de copias que componen el ciclo depende de varios factores: si los cambios se han realizado sobre la copia anterior así como su importancia, la seguridad de que no estén infectados, la política de la empresa o la costumbre que uno mismo tenga. De esta forma se puede controlar el alcance de la infección del virus, qué programas han sido infectados, a partir de qué fecha aproximadamente y con ello incluso cuál ha podido ser su posible procedencia.

Tener bajo control los discos de procedencia ajena

Es una práctica muy común el intercambio de *diskettes* con información y programas. Esta práctica supone uno de los riesgos más altos de contagio. Se debe mantener un severo control sobre los discos de procedencia ajena. La desinformación sobre el entorno de los virus puede suponer que la fuente más fiable de *software* se convierta en el principal foco contaminante.

En España, los programas piratas son la principal causa de contagio. Si se tiene como política no aceptar más *software* que el original, se está combatiendo a la vez las dos enfermedades informáticas actuales, los virus y la copia ilegal de programas.

Convertir las medidas preventivas en política interna de la empresa

En el seno de una empresa algunas de las medidas enumeradas anteriormente adquieren un valor especial. Las normas de prevención deben establecerse como política de la empresa y todos los empleados han de seguirlas forzosamente. Obligar a los empleados a utilizar *software* original y evitar el trasiego de *diskettes* de la oficina al domicilio particular, supone una disminución del riesgo de contagio.

Restringir el acceso directo a la información reservada

Otro foco peligroso en el entorno empresarial es el personal informático descontento, dado de baja o despedido. Antes de salir de la empresa ha podido introducir un "regalo" en los circuitos. Es conveniente controlar especialmente a estas personas en lo que se refiere a su acceso a información de valor para la empresa. Es fundamental restringir lo más posible el acceso directo a la información reservada.

Hasta ahora las medidas que se han tratado han sido de prevención para una de las mayores fuentes de virus, los soportes magnéticos. Pero existe un segundo entorno que resulta también vulnerable, las redes. El sistema informático español se encuentra conectado en su mayor parte por líneas punto a punto, que, según los técnicos, dificulta en gran medida la introducción de elementos externos. No se conoce aún ningún caso de contagio a través de las líneas telefónicas nacionales. Aun así se deben seguir unas medidas que disminuirán en gran parte el riesgo de contagio.

Comprobar los programas del BBS a los que se accede

Cuando se conecta con un BBS, se debe poner un especial cuidado en la utilización de los programas que éste posee. Deben comprobarse siempre estos programas.

El mismo método debe seguirse con todos los programas que se obtengan de cualquier tipo de red.
En los BBS, el riesgo de contagio es individual, por ser el propio usuario el que se pone en contacto a través del *modem*. Más difícil de solucionar es el caso de las redes que se pueden comunicar con sus asociados utilizando un correo interno de la red, problema que se trata a continuación.

Elegir convenientemente la clave de acceso a la red

La conexión con una red implica dos pasos obligatorios. El primero consiste en la introducción del número o clave de usuario, que tiene la misma composición que la del resto de usuarios y, por tanto, es de conocimiento público. El segundo consiste en la introducción de una palabra clave (*password*) propia de cada usuario. Una vez dentro de la red, cada usuario tiene un grado de privilegio dependiendo de su permiso de nivel de acceso.
La elección de una buena clave es de gran importancia, ya que es esencial para salvaguardar la seguridad de los datos y programas que contienen tanto el directorio del usuario, de forma directa, como el resto de directorios, de forma indirecta.
La clave de acceso debe seguir unas pequeñas reglas que dificultarán, sin llegar a evitarlo por completo, una posible intromisión. No debe tener ningún significado concreto y ha de ser sencilla, pues se debe memorizar para evitar que quede constancia de ella. Conviene cambiarla de vez en cuando, sin realizar un proceso rotativo. Se debe introducir en privado y no es conveniente utilizar claves típicas, como SESAMO, CHIP, nombres propios, nombres de mascotas, etc.
El hecho de que alguien entre de forma ilegal en una red a través de la localización de una de las claves, no sólo le da libertad para utilizar los datos de la cuenta a la que accede, sino que le permite la posibilidad de acceder a las cuentas de los demás usuarios.

Adoptar medidas preventivas en las empresas fabricantes de *software*

Existen unas pequeñas medidas preventivas que podrían aplicar los fabricantes de *software*. Su aplicación

es simple en comparación con las que deben tomar realmente las empresas como precaución, y ayudarían a evitar la venta de *diskettes* infectados.
Para empezar, se deben mantener en secreto los trabajos preventivos que se están llevando a cabo, para eliminar la posibilidad de que alguien cree un virus que vaya dirigido precisamente contra esos trabajos.
Resultaría de utilidad comercializar los *diskettes* sin espacio libre, para impedir así que se añadiese un código posterior y, por tanto, entrara un virus.
Las empresas deben depurar al máximo el contenido de los *diskettes* antes de comercializarlos. Deben realizar inspecciones periódicas. Resultaría práctico que el propio programa comprobara su estado antes de ejecutarse; por ejemplo, que con unos parámetros determinados del programa introducidos en un algoritmo se obtuviese un valor comparable en cada ejecución y que en caso de cambio se avisara al usuario.

No manejar virus

La última de las medidas preventivas resulta obvia: no se deben manejar virus. La creación de un virus aceptable implica un dominio excepcional de las posibilidades del ordenador y puede suponer para algunas personas un reto de superación. La manipulación de los virus puede acarrear consecuencias desagradables para el usuario. Este factor se incrementa cuando se intenta manipular un programa contaminante sin las herramientas apropiadas.

4.4. Medidas de detección

Comprobar cualquier *software* nuevo

Se debe comprobar cualquier *software* nuevo antes de considerarlo saludable y utilizable. No es conveniente fiarse plenamente de los programas originales, ni de los programas antivirus adquiridos por *shareware* y *freeware*.
Ejemplos de un posible contagio por obviar esta medida son la ya mencionada revista que distribuyó en nuestro país el virus Viernes-13 en sus programas originales, y el antivirus Flushot, de dominio público, que contie-

ne un programa contaminante en una de sus versiones. Una de las pruebas consiste en mantener el nuevo programa adquirido en observación, durante un tiempo prudencial, en un ordenador de cuarentena. Todos los programas y *diskettes* pasarán por este ordenador durante el tiempo estimado que permita observar si el nuevo *software* puede ser dañino, tanto en el funcionamiento del ordenador como en la información almacenada.
Examinar el nuevo programa implica no introducir ningún otro *diskette* con información importante, ni manipular el disco duro hasta que no se haya apagado completamente el ordenador una vez probado el programa. Es posible que con un solo examen se pasen por alto algunos detalles importantes y que en una segunda prueba se vean con más claridad.

Controlar los cambios de tamaño en los ficheros ejecutables

Se debe mantener un control de los cambios de tamaño en los ficheros ejecutables. Como se ha visto en capítulos anteriores, algunos virus se adhieren a programas ejecutables, lo que implica que el parámetro que marca el tamaño de dicho programa aumentará de valor al producirse la infección del fichero.
Hay virus que una vez que han infectado el programa no vuelven a introducirse en él, pero otros sí lo hacen, de manera que aumenta el tamaño del programa repetidas veces hasta no poder ejecutarse por no tener suficiente memoria. Esto ocurría en los ficheros .EXE con el Viernes-13 en su primera versión.
Antes de ejecutar un programa se debe hacer una consulta de su tamaño y fecha de compilación para compararlo con las copias que se tengan almacenadas en los *backups* y originales, así como de sus atributos una vez terminada la ejecución. Para poder realizar estas comparaciones es conveniente crear un fichero particular para cada programa en el que se tenga el tamaño, la fecha y la hora de su última modificación.

Inspeccionar periódicamente el soporte magnético y la memoria libre en el disco

Es aconsejable realizar inspecciones periódicas del soporte magnético para comprobar si posee sectores

defectuosos (*bad clusters*), y del tamaño de la memoria libre que existe en el disco. Los parámetros de estas dos medidas son insignificantes pero pueden llegar a delatar la presencia del intruso. Para llevar a cabo esta operación se puede utilizar un programa de utilidades de disco, como las Utilidades Norton, que ofrecen mapas de la utilización de la memoria tanto primaria como secundaria.

Observar síntomas de infección vírica

De la misma forma que los síntomas humanos de una gripe son la tos y la fiebre, existen ciertos síntomas de infección de un ordenador. Estos factores son: mayor lentitud en el proceso, mayor tiempo de respuesta, pérdida de espacio en la memoria, falta de espacio para la ejecución en la memoria de trabajo en programas que antes sí se podían ejecutar, anomalías en la pantalla, resultados inesperados en la ejecución de programas, etc. Se debe observar con especial atención si en la ejecución de un programa o comando de sistema se produce alguna operación inesperada de entrada/salida sobre el soporte magnético. La medida adquiere especial interés cuando estas operaciones tardan excesivamente en realizarse.

Instalar *software* de seguridad

Es conveniente instalar *software* de seguridad. Por un lado, los detectores y protectores ayudarán a delatar el virus en su período de incubación, antes de causar el daño para el que fue creado. Los programas antivirus, vacunas y antídotos, por otro lado, permiten actuar sobre el propio virus y sobre los daños que ha cometido, en el caso de ser los apropiados para combatir ese virus. Si en todos los discos de arranque se introduce un detector que sea llamado por el programa AUTOEXEC.BAT, éste comprobará los programas del sistema, objetivo principal de virus como el (c) Brain y el famoso virus de la pelotita, Italian.

4.5. Planes de contingencia

El plan de contingencia incluye las medidas que se deben seguir en el caso de detectar la presencia de un

virus. Estas medidas desbancan a las de prevención desde el momento en que, a través de un programa detector o como resolución de una de las medidas de detección anteriormente detalladas, se haya encontrado el virus en su período de latencia o reproducción.

En cualquiera de los casos, los pasos a seguir son los mismos, siempre y cuando no se disponga de un programa antivirus que ahorre alguno de ellos, como los referentes a limpieza del disco, cuyo trabajo es más laborioso. Son universales, es decir, se deben seguir independientemente del virus que haya infectado el sistema y en el mismo orden en que aparecen.

Conservar la calma

De la misma forma que la primera de las medidas preventivas era no temer a los virus, el primero de estos pasos es conservar la calma.

Desconectar totalmente el ordenador

Se debe desconectar el ordenador con el interruptor o desenchufando. No hay que conformarse con reinicializar el ordenador mediante la combinación de las teclas **<Ctrl-Alt-Supr>**. Existen virus que bloquean el teclado plenamente y puede llegarse a dar el caso de que el programa contaminante simule una desconexión permaneciendo en la memoria.

Encender el ordenador con otra copia de seguridad

Se encenderá de nuevo el ordenador utilizando un sistema operativo original o una copia realizada con un sistema sano. Se debe tomar la especial precaución de no volver a utilizar la copia habitual del sistema de arranque por si está contagiada.

Hacer copias de seguridad de los ficheros de datos y ficheros no ejecutables

Habrá que realizar copias de seguridad, únicamente de los ficheros de datos y de todos aquellos programas que no sean ejecutables, ya que sabemos que son los que ocultan el virus entre sus instrucciones.

Dar nuevo formato al disco infectado

Se debe dar nuevo formato al disco infectado, salvo en el caso en que no se tengan copias originales de los

programas que lo integran y se intente anular el virus sobre el mismo programa o *diskette*. La excepción de no dar formato al soporte magnético se deberá llevar a cabo si se trata de algún tipo conocido de virus y se sabe cómo actúa, dónde se camufla, cómo desactivarlo, o si se va a adquirir un programa antivirus que lo desactive.

Comprobar los discos originales con programas detectores

Si no se posee un programa detector, se debe buscar en el programa original la misma señal que llevó a confirmar la existencia de contagio. En caso afirmativo se deberá volver al primero de estos pasos, siendo imprescindible comprar los programas de nuevo.

Recomponer el disco con los programas originales

Este paso considera negativa la prueba realizada en el paso anterior con los discos originales.

Volver a copiar la información no infectada

Se debe volver a copiar toda aquella información que se sustrajo en un principio por encontrarse al margen de la infección. Esto incluye los ficheros de datos de los que se hizo una copia de seguridad.

Volver a comprobar los discos con programas detectores

Una vez recompuesto el disco con la estructura que tenía anteriormente, se debe pasar de nuevo un detector buscando posibles restos del virus.

Probar todos los discos utilizados con anterioridad a la infección

Este paso resulta conveniente, ya que es muy posible que se detecte tarde la infección y se haya propagado a más *diskettes*.

Avisar a otros usuarios del peligro de infección

Comunicar la posibilidad de infección a todas las personas con las que se haya tenido intercambio de información, y si se está conectado a alguna red, comunicar del peligro de contagio a su administrador.

4.6. Programas antivirus, vacunas y detectores

Existe infinidad de programas en el mercado que permiten proteger el ordenador y reparar los daños producidos por la acción de los virus. Estos programas se pueden clasificar por las funciones que realizan: prevención, detección, vacunación, identificación y control de daños.

Estas funciones se solapan en la mayoría de estos programas. Cada programa quedará enmarcado en la categoría que indique su principal función. Por tanto, no se puede hablar de programas protectores, detectores, vacunales y restauradores propiamente dichos.

A continuación se detallan las funciones reseñadas anteriormente.

4.6.1. Programas de prevención

Los programas de prevención impiden que los virus infecten el ordenador en un primer momento. Detectan y rechazan cambios en los ficheros ejecutables, evitando que el virus incluya códigos perniciosos dentro de ellos. Algunos no permiten que los programas permanezcan residentes en memoria si no están en una lista de aplicaciones autorizadas y comprobadas previamente. La mayoría de los programas preventivos utilizan esta última característica, ya que estos productos son los que dan mejor resultado.

4.6.2. Programas de detección

Los programas de detección avisan de una posible infección, buscando los síntomas característicos de cada virus. Estos programas comprueban los ficheros ejecutables, el sector de arranque de los discos (*boot*), etc. y los comparan con una clave previamente grabada. Esta clave se crea pasando ciertos bytes del fichero por un algoritmo, generalmente secreto, que genera un número. Esta técnica se denomina suma de verificación. La probabilidad de que alguien varíe el programa sin que se produzca un cambio en la clave es infinitesimal. Algunos pro-

ductos comprueban las claves de todos los ficheros a la vez, mientras que otros comprueban la clave de cada programa cuando se ejecuta.

4.6.3. Programas de vacunación

Un programa que vacuna se "autoinyecta" en cada programa, es decir, es como un virus benigno. Cuando se ejecuta un programa vacunado, el código inyectado realiza una comprobación y avisa si se produce algún cambio.

4.6.4. Programas de identificación

Algunos programas antivirus intentan identificar virus conocidos, por medio de la localización de determinadas cadenas de caracteres ASCII dentro del código de los programas. Como ejemplo de estas cadenas se puede citar "SUMSDOS", cadena identificativa de las primeras versiones del virus Viernes-13.

4.6.5. Control de daños

Cualquier programa antivirus que se precie debe ofrecer un control de los daños producidos, tanto en el aspecto preventivo como restaurador. Las técnicas preventivas incluyen la detección de intentos de acceso directo al disco, avisando de los programas que intentan realizar este tipo de operación sin autorización. Protegen contra escritura el disco duro mientras que comprueban el *software* de dudosa procedencia. Si lo peor ocurre y el virus borra los programas o formatea el disco, se podrá restaurar el daño si el programa antivirus provee de alguna utilidad de recuperación de ficheros o conserva una copia de la FAT.

4.6.6. Funciones complementarias

A la par de estas medidas comunes de protección, algunos programas antivirus ofrecen características de ayuda adicional. Si el programa salva una copia de la

memoria CMOS, se podrá usar para restaurar la configuración cuando las pilas del ordenador se gasten. Cuando accidentalmente se intenta copiar un fichero ejecutable sobre otro con el mismo nombre, dicho programa antivirus avisará.

4.6.7. Inconvenientes

Todas las funciones anteriores presentan algún inconveniente en su uso. Los programas de prevención requieren una costosa preparación antes de poder utilizarse correctamente. La lista de programas autorizados debe ser actualizada constantemente para evitar que los nuevos programas adquiridos sean rechazados por el antivirus.

Los programas detectores son muy especiales y sólo suelen servir para un virus en particular. Una pequeña variación en el virus puede hacer que el detector quede obsoleto.

Los programas de identificación no son excesivamente prácticos, ya que el más leve cambio en la cadena de caracteres del virus puede anular la función del identificador.

Por último, aunque el programa antivirus tenga un control de daños, si el virus ha realizado un formateo a bajo nivel del disco, le será imposible recuperar la información almacenada.

5

El marco legal

5.1. La legislación en Estados Unidos

A nadie se le escapa el adelanto que Estados Unidos representa en materia informática con respecto a nuestro país. Esta situación de vanguardia le supone, sin embargo, una situación de entorno perfecto para la proliferación del delito informático. A pesar del adelanto tecnológico, el derecho relativo a este tipo de delitos no ha evolucionado con la sociedad. Así la legislación americana se ha visto obligada a modificar apresuradamente las leyes existentes o a crear leyes nuevas que protejan la información y castiguen al infractor de delitos contra la propiedad informática.

Actualmente, la legislación de Estados Unidos recoge la Ley contra el Abuso y Fraude Informático, creada en 1984 y modificada en 1986. Particularmente, algunos estados han ido adecuando sus leyes a los delitos informáticos que se han ido produciendo.

Uno de los precursores en la actualización de su legislación fue el estado de California. A finales de 1988 modificó su código penal, haciendo constar que todo aquel que "conscientemente acceda y, sin permiso, añada, altere, erosione, borre o destruya datos, *software*, o programas de ordenador, sistema informático o red de ordenadores...", es culpable de ofensa pública. La pena que se estipula para este delito de ofensa pública supone una multa de 10 000 dólares, la confiscación del equipo informático del acusado y prisión hasta un período máximo de tres años.

El Ministerio de Justicia norteamericano ha definido el delito informático de forma concisa. Delito informático es cualquier acto ilegal para el que es esencial el conocimiento de la tecnología informática para su comisión, investigación o persecución. También describe algunas de las técnicas más frecuentemente utilizadas en la comisión de tales delitos, como por ejemplo el caballo de Troya, la bomba lógica, la técnica del salami, la sustracción de información, la filtración de datos, el espionaje industrial, etc.

La técnica del salami consiste en introducir o modificar instrucciones en los programas para reducir sistemáticamente en unos céntimos las cuentas corrientes, los saldos de proveedores, etc. Estos céntimos son transferidos a una cuenta que se abre con nombre supuesto bajo el control del defraudador.

El primer caso de juicio por delito informático se celebró a finales de noviembre de 1988 en el estado de Texas. El tribunal de Forth Worth declaró culpable a Donald Burleson, americano de cuarenta años, de haber propagado un virus en un ordenador.

A Burleson se le acusaba de haber saboteado en 1985 la información de una compañía de seguros en la que había estado trabajando. La acusación, definida como "acceso dañino o perjudicial a un ordenador le supuso al acusado una pena condenatoria de dos años de prisión y una indemnización de cinco mil dólares. Previamente, Donald Burleson, había sido condenado, a título civil por daños y perjuicios, a pagar doce mil dólares a su antigua compañía.

El fiscal argumentó que el acusado, de profesión programador, había introducido un virus que se disimulaba en un programa de apariencia normal. El programa, con características de bomba lógica, se activaba después de una secuencia de órdenes habituales del sistema. Con este retardo en la activación, el autor conseguía una buena coartada, para el momento en que la mano inocente provocase la catástrofe.

La defensa, por su parte, argumentó la inocencia del acusado, manteniendo que alguien se había servido de la clave de acceso al ordenador de su cliente para realizar la fechoría en su perjuicio.

Otro caso ha reclamado la atención mundial al final de

la década, es el juicio contra Robert Tappan Morris. El ocho de enero de 1990 comenzó, en el tribunal del distrito de Siracusa, Nueva York, la vista contra el estudiante acusado de introducir un gusano informático en la red Internet y conseguir su paralización en noviembre de 1988. Se espera que este juicio siente jurisprudencia, ya que por primera vez se encuentra en el banquillo de los acusados alguien responsable de extender una infección en una red pública.

La defensa argumentó que su cliente era un claro defensor de la mayor seguridad de los sistemas informáticos. Para apoyar su tesis presentó un vídeo que recogía una conferencia pronunciada por Morris en 1987 ante expertos de la Agencia nacional de seguridad, sobre los medios de acceso ilegal a ordenadores; circunstancia que la fiscalía anunció que utilizaría contra el acusado, cumpliendo posteriormente su afirmación.

Por su parte, Morris se declaró inocente basándose en que la intrusión ilegal en la red la había realizado únicamente como medio de prueba para demostrar la inseguridad de la instalación. Además declaró que el programa que había colocado en la red era benigno. Debe entenderse la benignidad, en este caso, como el hecho de no haber destruido ningún tipo de información. Queda clara la escasa benignidad de un programa que en cuestión de pocas horas consiguió paralizar una red de más de 6000 ordenadores.

El padre del acusado, funcionario del estado americano, y experto en seguridad de sistemas informáticos, no fue llamado a declarar para evitar que con ello se pudiesen desvelar detalles de secretos militares.

La fiscalía solicitó una acusación de culpabilidad y una pena de cinco años de cárcel, además de una multa de un cuarto de millón de dólares para este personaje, héroe para unos y terrorista informático para otros.

Robert Tappan Morris fue declarado culpable de "irrumpir intencionadamente en ordenadores federales sin autorización y alterar, dañar o destruir información". La sentencia, fechada el cuatro de mayo de 1990, condenaba a Morris a tres años de libertad condicional, 10000 dólares y el cumplimiento de 400 horas de trabajos comunitarios. El magistrado no acogió los argumentos de los fiscales, que solicitaban la pena máxima.

5.2. La legislación en España: La Ley de la Propiedad Intelectual

En España todavía no se goza de una ley que condene los delitos de fraude o abuso informático de una manera directa. Independientemente de las acciones derivadas de las infracciones habituales al código civil y al código penal, la única forma de atacar este tipo de delito está regulada en la Ley de la Propiedad Intelectual (L.P.I.).

La L.P.I. está recogida en el Boletín Oficial del Estado número 275, del 17 de noviembre de 1987. Concretamente, se hace referencia en su título VII, "De los Programas de Ordenador", que contiene los artículos 95 a 100 de la citada Ley. No supone un marco jurídico idóneo para enjuiciar a delincuentes relacionados con delitos de la información en soporte magnético, ya que se trata de una ley poco concisa y que deja escapar la mayoría de los detalles a programas con fines perniciosos.

A continuación se da un repaso a los artículos de la Ley que hacen referencia a los delitos de ordenador cometidos. En su primer artículo la Ley reza como sigue: "La propiedad intelectual de una obra literaria, artística o *científica*, corresponde al autor por el solo hecho de su creación". Este artículo centra el primer paso para interpretar un enjuiciamiento de un presunto invasor de un sistema por medio de un programa vírico o similar. Se trata de considerar al infractor como autor de una propiedad intelectual científica, su programa contaminador, de la que como autor por esta Ley es responsable, ya que el primer artículo le otorga los derechos y, por tanto, los deberes de autor.

Por otro lado, en su artículo 96, la L.P.I. se encarga de definir exactamente lo que se entenderá como programa de ordenador en la aplicación de la Ley: "A los efectos de la presente Ley, se entenderá por programa de ordenador toda secuencia de instrucciones o indicaciones, destinadas a ser utilizadas directa o indirectamente en un sistema informático para realizar una función o tarea, o para obtener un resultado determinado, cualquiera que fuera su forma, expresión y fijación".

Analizando un poco al detalle el documento, se puede extraer el ámbito de protección del *copyright*, de esta Ley.

Este se puede resumir en los siguientes puntos:

- Programa fuente y objeto.
- Documentación técnica y manuales de uso.
- Versiones sucesivas.
- Programas derivados.
- Respecto a los programas que forman parte de una patente o de un modelo de utilidad, gozarán de lo dispuesto en la Ley, de la protección que pudiera corresponderle por aplicación del régimen jurídico de la propiedad industrial.

Empieza a perfilarse, a la vista de los datos, que la acusación se debe enfocar por parte de los afectados, en la modificación de sus programas fuente con la introducción de los virus. Téngase en cuenta que generalmente los *copyrights* de los programas están en posesión de las casas comerciales, y son ellas quienes deberían encabezar la guerra contra los promotores de virus. Esta idea es una solución provisional, a la espera de una necesaria ampliación de la Ley.

Las acciones que pueden derivarse contra los infractores de los derechos de autor deben enfocarse desde dos puntos de vista: la demanda civil y, si procede, la querella o denuncia. La Ley Orgánica 6/87 modificó el artículo 534 del código penal en alguno de sus puntos. En ellos se establece multa de 30 000 hasta 600 000 pesetas en casos de reproducción, distribución o divulgación; arresto mayor y multa de 50 000 a 1 500 000 pesetas con circunstancias agravantes, como ánimo de lucro, infracción del derecho de divulgación del autor, usurpación de la condición del autor o modificación sustancial de la integridad de la obra; prisión, multa de 50 000 a 3 000 000 de pesetas y posible cierre de la empresa infractora cuando la cantidad o el valor de las copias ilícitas posean especial trascendencia económica, cuando el daño causado revista especial gravedad.

La primera denuncia por un delito de este tipo se debió a la publicación informática que divulgó involuntariamente el virus Viernes-13 con su *diskette*. Es un caso atípico, puesto que no se denunciaba a personas concretas, sino que se hacía referencia a un acto de sabotaje.

Por otro lado, la primera denuncia contra los presuntos

autores de un virus se presentó en el juzgado número 8 de Barcelona, en mayo de 1989. La denuncia fue presentada conjuntamente por la empresa en la que trabajaban los programadores denunciados y la Asociación de Empresas Españolas Fabricantes de Software.

Los hechos, según reza la denuncia, ocurrieron como sigue. La citada empresa se dedicaba a desarrollar y comercializar programas de gestión. Sus principales clientes eran los ayuntamientos españoles. Uno de los presuntos maleantes aprovechó su puesto de responsabilidad en la empresa para enviar a los clientes un *diskette* con supuestas instrucciones para corrección de defectos del programa comercializado. Al poco tiempo los ayuntamientos se encontraron con un virus en la memoria de su ordenador.

Poco tiempo después, uno de los denunciados abandonó la empresa llevándose programas fuente, rutinas de programación y listados, sin recoger el finiquito, mientras que el otro había sido previamente despedido. Ambos montaron su propia oficina de mantenimiento y se presentaron en varios ayuntamientos afectados para ofrecer sus servicios asegurando que sabían cómo acabar con el virus. Ambos programadores se declararon inocentes y el caso está visto para sentencia.

5.3. Cuando las leyes no resarcen: seguro informático

Ante la insuficiente protección que ofrece la legislación española se hace necesario otro tipo de medidas para proteger la información.

Esta oportunidad ha sido aprovechada por alguna empresa de mantenimiento de equipos informáticos y alguna compañía de seguros. El asunto consiste en desarrollar un servicio que cubre a las empresas ante siniestros accidentales o provocados en la información, con forma de póliza de seguros.

Consiste en establecer un análisis previo de los sistemas y equipos informáticos y asegurar la instalación, puesta a punto y soporte de los equipos. No es la solución ideal, pero puede resarcir al usuario de la pérdida de su información con una compensación económica.

Apéndice A

Repaso de las nociones básicas del DOS

A.1. Estructura del DOS

Para una mayor comprensión del funcionamiento de los virus y de las medidas que se pueden adoptar contra ellos es conveniente refrescar algunos conceptos.

En primer lugar se podría definir un sistema operativo como el *software* (soporte lógico o conjunto de programas) que controla el *hardware* (soporte físico u ordenador). Esta definición, válida hace algunos años, queda obsoleta ante la tendencia actual de asignar funciones del *software* a la memoria fija del ordenador. Hoy día, las funciones del *hardware* superan a las del *software*.

Por tanto, se hace necesaria una nueva definición del sistema operativo: Es una serie de programas, dispuestos en el *software* o en memoria fija, que hacen que el *hardware* sea utilizable. El *hardware* suministra funciones básicas de operación y programación. Los sistemas operativos ponen estas funciones a disposición del usuario.

El DOS o MS-DOS (*MicroSoft Disk Operating System*) es el sistema operativo de los ordenadores personales (PC) más extendido y utilizado. Por ello, la mayoría de los piratas informáticos o *hackers* han desarrollado su labor en este entorno.

La estructura del DOS se compone de varias capas que aíslan el núcleo (*kernel*) del sistema operativo, del usuario. Estas son: el intérprete de comandos o mandatos (*shell* o COMMAND.COM), el BIOS (*Basic I/O System*, sistema básico de entrada y salida) y el núcleo del DOS.

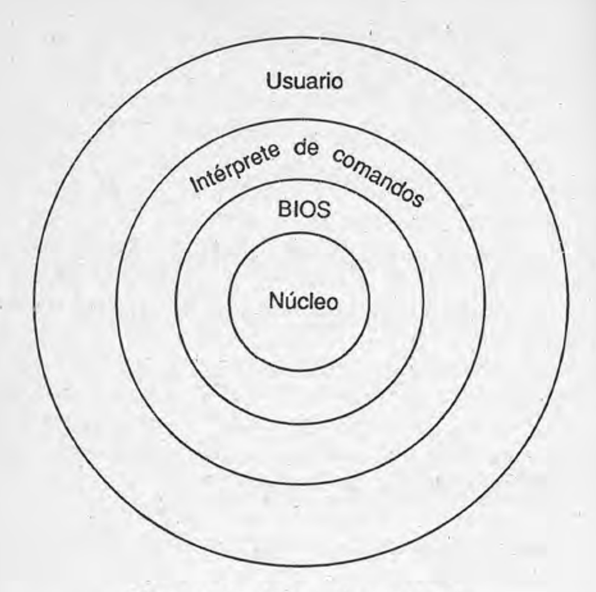


Figura A.1. *Estructura del DOS*

A.1.1. El intérprete de comandos

El intérprete de comandos del sistema operativo, también llamado *shell* (capa), es el interfaz del usuario con el sistema operativo, es decir, es el entorno que permite al usuario comunicarse con el DOS. Su función consiste en analizar gramaticalmente y gestionar los comandos que el usuario envía al sistema operativo. Por ejemplo, cuando se escribe el comando DIR, el intérprete de comandos comprueba su sintaxis, y a continuación realiza las funciones necesarias para ejecutarlo.

En el sistema operativo DOS, el intérprete de comandos por defecto es el programa COMMAND.COM, que se encuentra en el directorio raíz de los discos que disponen de sistema operativo. Si el usuario es un programador experto, puede cambiar el intérprete de comandos por uno propio, añadiéndolo al fichero de configuración (CONFIG.SYS) del sistema dentro del disco de arranque del ordenador.

El COMMAND.COM es el responsable de la gran

mayoría de los mensajes y respuestas que da el sistema operativo, que constituyen el grueso de las informaciones emitidas y recibidas por el usuario. Aunque sea la parte más visible del DOS y la que más comunicación tiene con el usuario, hay que hacer notar que el intérprete de comandos no es el sistema operativo, sino un programa que se ejecuta bajo el control del DOS.

El intérprete de comandos se estructura en tres partes: una parte residente en memoria, otra sección de inicialización y un módulo transitorio.

La parte residente en memoria contiene las subrutinas que procesan los comandos de ruptura **<Ctrl-C>** y **<Ctrl-Inter>**, los errores críticos y la terminación normal de los programas que ejecuta el usuario. Es también el responsable de los mensajes de error que aparecen en la pantalla del ordenador cuando se ha escrito un comando incorrectamente o surge algún problema de ejecución. Esta parte contiene, además, un programa o código que carga la parte transitoria del intérprete cuando sea necesario.

La sección de inicialización es la encargada de procesar el programa *batch* AUTOEXEC.BAT cuando existe en el directorio raíz del disco de arranque. Una vez realizada esta función, se desecha.

La parte transitoria del intérprete de comandos es la responsable de construir y sacar en pantalla el indicador de comandos o *prompt* del sistema (símbolo que aparece en la pantalla del ordenador, que nos indica dónde debemos introducir el comando siguiente). Puede estar diseñado por el usuario con el comando PROMPT. Si el usuario no ha diseñado ningún *prompt*, la parte transitoria del COMMAND.COM produce uno por defecto. También se encarga de leer las órdenes o comandos desde el teclado o desde los ficheros *batch* (ficheros con extensión .BAT), así como de su posterior ejecución.

Cuando termina de ejecutarse un programa, la porción residente del COMMAND.COM realiza una comprobación de la parte transitoria para saber si ha sido destruida por el último programa ejecutado. En el caso de haber sido destruida vuelve a cargar una nueva copia.

Los comandos del sistema operativo y de usuario que puede procesar el COMMAND.COM se dividen en tres clases: comandos internos, comandos externos y ficheros *batch*.

Los comandos internos están implementados en el propio COMMAND.COM y son, entre otros, DEL, COPY, DIR, CLS, etc.

Los comandos externos o programas transitorios son nombres de comandos y programas que están almacenados en ficheros de disco. Al escribir el nombre de un comando externo, el intérprete de comandos realiza una serie de funciones. Primero debe buscar el programa en el disco recorriendo la vía de acceso indicada por el comando PATH; si lo encuentra, tiene que cargar el fichero desde el disco a la memoria transitoria de programas (TPA, *Transitory Program Memory*), dar el control al programa que se va a ejecutar y una vez que éste finaliza debe borrar la memoria. Si se quiere volver a ejecutar el mismo comando externo, éste debe ser cargado otra vez desde disco a la memoria.

Los ficheros *batch* son ficheros de texto que contienen una serie de comandos, ya sean internos o externos, que serán ejecutados secuencialmente cuando se llame al fichero *batch*. El COMMAND.COM contiene un intérprete especial, situado en la parte transitoria del mismo, que procesa este tipo de ficheros. El intérprete lee línea a línea los comandos incluidos en el fichero *batch* y los va ejecutando uno por uno.

Los pasos que sigue el intérprete de comandos al intentar ejecutar una orden del usuario son los siguientes:

1. Comprueba si el comando es interno o externo.
2. Busca el comando externo y los ficheros *batch* en el directorio de trabajo y en los directorios o discos especificados en la instrucción PATH. Busca primero los programas o comandos con extensión .COM, después los de extensión .EXE y, por último, los que tienen extensión .BAT.
3. Si encuentra el fichero, comando o programa solicitado por el usuario, el intérprete de comandos llama a la función EXEC (ejecutar programa) del sistema operativo para cargarlo en memoria y ejecutarlo.

Un ordenador desconectado no contiene sistema operativo alguno, puesto que éste se encuentra en dispositivos de almacenamiento externo (discos). Por tanto, al encender el ordenador o reinicializarlo (bien pulsando el

botón de RESET o pulsando simultáneamente las teclas <Ctrl-Alt-Supr>), el DOS debe ser cargado en memoria. El proceso de carga del sistema operativo es el siguiente:

1. Al encender el ordenador se ejecuta un programa situado en la memoria ROM (*Read Only Memory*, memoria de sólo lectura) que llama al programa de comprobación del sistema y a la subrutina de arranque de la ROM (ésta función es dependiente del *hardware*).
2. Esta subrutina, llamada *bootstrap*, lee el programa de arranque del disco, que está situado en el primer sector del mismo, llamado sector de arranque o *boot track*, lo carga en memoria y le transfiere el control.
3. Este último programa carga el sistema operativo y le da el control al intérprete de comandos.

A.1.2. El BIOS

El módulo BIOS es específico de cada sistema *hardware* y depende del fabricante del ordenador. Por lo general, hay pocas variaciones en los ordenadores personales IBM y sus clónicos. Contiene las unidades dependientes del *hardware* que están residentes por defecto dentro de los dispositivos, como el teclado (en DOS representado por la abreviatura CON de consola), la impresora (representado por la abreviatura PRN de *printer*, impresora en inglés), dispositivo auxiliar (representado por AUX, abreviatura de auxiliar), hora y fecha (representado por CLOCK, reloj en inglés) y el dispositivo para el disco de arranque (dispositivo de bloques).

El núcleo del DOS se comunica con estas unidades de control por medio de paquetes de entrada/salida (E/S); estas unidades traducen las peticiones realizadas por el sistema, transformándolas en las órdenes necesarias para que funcionen los distintos controladores del *hardware*.

El BIOS se almacena en la memoria RAM (*Random Access Memory*, memoria de acceso aleatorio), durante la inicialización del sistema, formando parte de un fichero llamado IO.SYS o IBMBIO.COM (en los ordenadores IBM y clónicos). Este fichero tiene los atributos oculto (*hidden*) y de sistema (*system*), que hacen que cuando se ejecuta el

comando DIR del DOS en el directorio raíz, permanezca oculto al usuario.

A.1.3. El núcleo del DOS

El núcleo del DOS es el encargado de gestionar el DOS, de tal forma que los programas de aplicación puedan utilizar las funciones o servicios del sistema operativo. Es un programa que contiene un conjunto de servicios independientes del soporte físico del ordenador, que se llaman funciones del sistema. Estas funciones son: la gestión de ficheros y registros, la gestión de la memoria, la gestión de los dispositivos de entrada y salida de caracteres, la generación de otros programas y la gestión de acceso al reloj de tiempo real.

Para que un programa pueda utilizar las funciones del sistema, debe cargar en los registros del microprocesador los parámetros específicos de la función que quiere utilizar. A continuación, debe ceder el control al sistema operativo por medio de una llamada o interrupción *software*.

El núcleo del DOS se carga en la memoria del ordenador al inicializar el sistema. Está contenido en el fichero MSDOS.SYS o IBMDOS.COM (en los ordenadores IBM y clónicos). Este fichero también tiene los atributos oculto y de sistema.

A.2. La memoria

La memoria del ordenador se compone de pequeñas unidades de almacenamiento denominadas bits (contracción de las palabras inglesas *binary digit*). Los bits tienen dos estados posibles: 0, o desactivado, y 1, o activado; es decir, sirven para almacenar números binarios a razón de un dígito por bit. Por consiguiente, toda información almacenada en el ordenador se guarda en forma de series de unos y ceros.

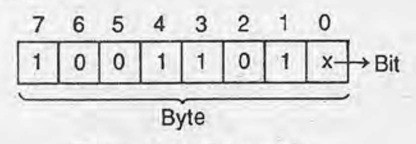


Figura A.2. *Byte y bit*

Los bits se agrupan de 8 en 8, formando así bytes u octetos. Un byte puede ser direccionado, es decir, se puede acceder a él mediante un número que es su dirección específica.

La cantidad de memoria de un ordenador se mide en kilobytes (1 Kb=1024 bytes). Actualmente, debido al avance de los ordenadores, la memoria puede medirse en megabytes (1 Mb=1024 Kb) y llegará un momento en que deberá medirse en gigabytes (1 Gb=1024 Mb).

El direccionamiento de la memoria del ordenador se realiza con dos bytes (una palabra). Con una palabra se pueden direccionar 256x256 bytes o, lo que es lo mismo, 65536 bytes o 64 Kb. Para superar este valor se utiliza una técnica conocida como segmentación. Con la segmentación se utilizan dos registros: uno llamado segmento y otro denominado desplazamiento, *offset* o dirección relativa. Luego una dirección de memoria tiene dos componentes y se expresa de la forma "segmento:desplazamiento" (véase figura A.3). Por cada segmento se podrán direccionar 65536 posiciones de memoria (de la 0 a la 65535). La numeración de las posiciones de memoria empezará en el segmento 0, desplazamiento 0 (0000H:0000H) y terminará en el segmento 65535, desplazamiento 65535 (FFFFH:FFFFH). Con la técnica de la segmentación se podrían direccionar teóricamente hasta 65536x65536 bytes de memoria.

Los ordenadores personales utilizan dos tipos de memoria: la memoria ROM (*Read Only Memory*), de sólo lectura, y la memoria RAM (*Random Access Memory*), de acceso aleatorio de lectura y escritura.

El espacio direccionable por los ordenadores personales es de 1 Mb. Este espacio direccionable o mapa de memoria se refleja en la tabla A.1.

A.3. Formato y organización de los discos

Los discos son dispositivos magnéticos de almacenamiento permanente de la información. Se presentan en varios tamaños y capacidades pero operan básicamente de la misma forma. Como dispositivos físicos tienen una

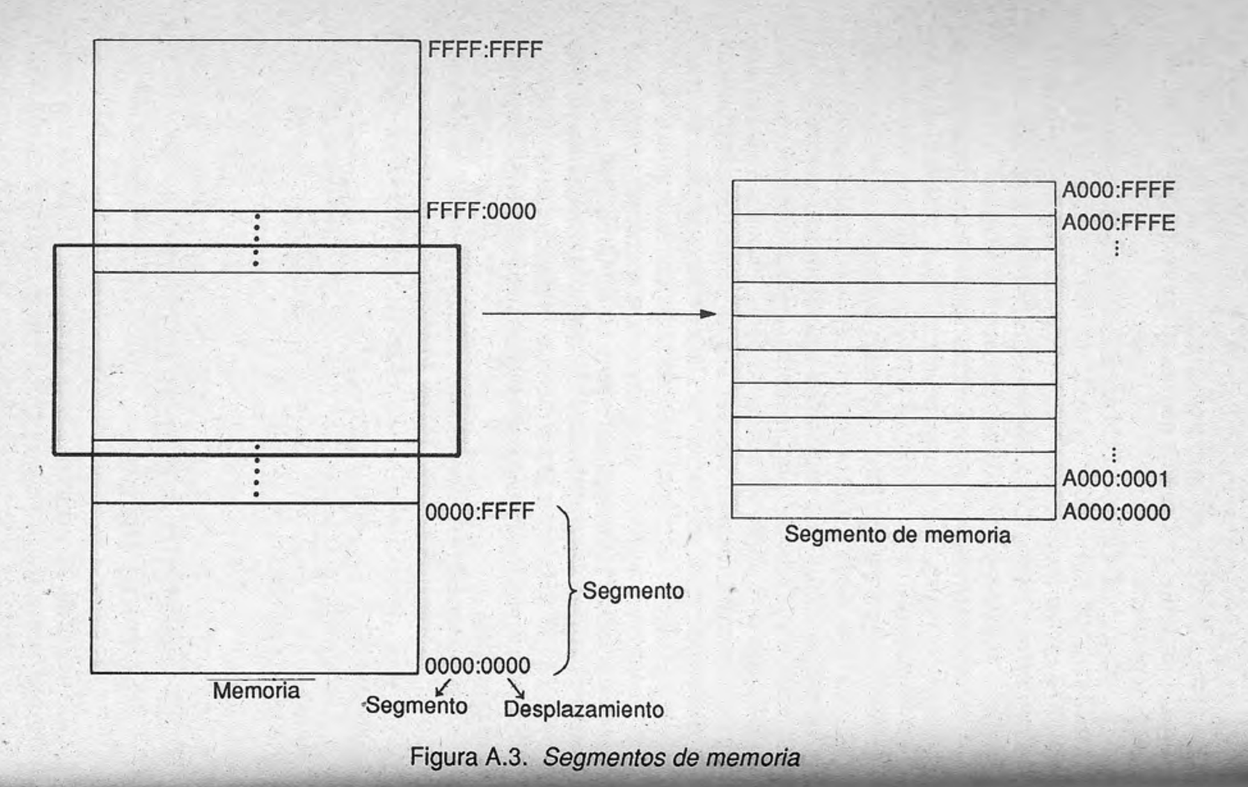


Figura A.3. *Segmentos de memoria*

Tabla A.1

Espacio direccionable (Kb)	*Longitud* (Kb)	*Direcciones* (Hex.)	*Contenido*
0	0,125	0 - 7F	Vectores de interrupción del BIOS.
0,125	0,125	80 - FF	Vectores de interrupción del DOS.
0,25	0,25	100 - 1FF	Vectores de interrupción del usuario.
0,5	0,5	200 - 3FF	Vectores de interrupción del BASIC.
1	0,25	400 - 4FF	Area de datos del BIOS.
1,25	0,25	500 - 5FF	Area de datos del DOS.
1,5	62,5	600 - FFFF	Memoria del usuario.
64	192	10000 - 2FFFF	Area de expansión de la memoria.
256	384	30000 - 9FFFF	Area de expansión de la memoria.
640	64	A0000 - AFFFF	Memoria de expansión de la pantalla.
704	64	B0000 - BFFFF	Area de memoria de la pantalla.
768	192	C0000 - CFFFF	Area de ROM: expansión.
960	16	F0000 - F3FFF	Area de ROM reservada.
976	8	F4000 - F5FFF	Area de ROM del usuario.
984	32	F6000 - FDFFF	Area de ROM: BASIC.
1016	8	FE000 - FFFFF	Area de ROM: BIOS.

estructura física y como dispositivos de almacenamiento de información tienen un formato lógico.

Los discos se estructuran físicamente en círculos concéntricos, llamados pistas. Cada pista se divide en segmentos de igual tamaño, llamados sectores. La cantidad de información que se puede almacenar en un disco depende de su capacidad, es decir, del número de pistas que contenga y del tamaño de sus sectores, y la capacidad depende de la densidad, que es el número de pistas por pulgada.

Con el desarrollo de los discos duros se creó un nuevo concepto, el de cilindro. Un disco duro se compone de varios discos situados en planos paralelos entre sí y con el mismo eje. La misma pista de cada uno de los discos constituye un cilindro.

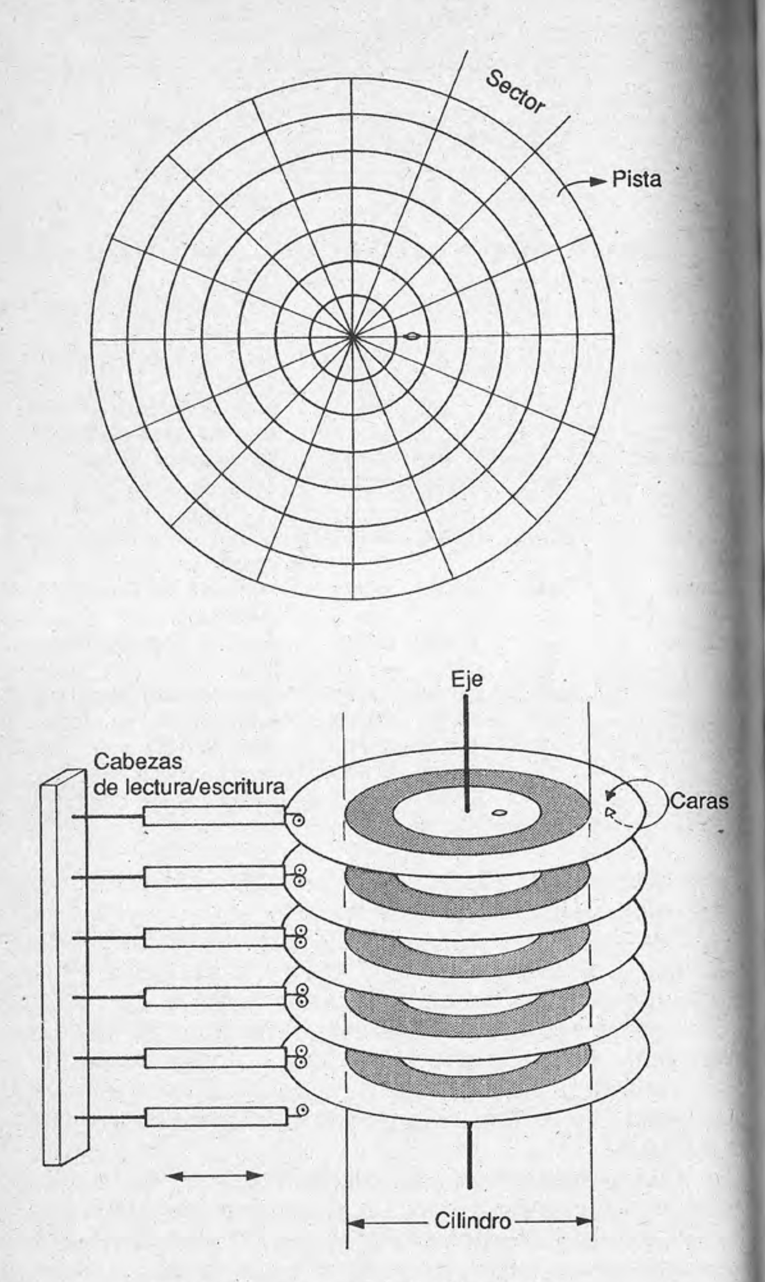


Figura A.4. *Sectores, pistas y cilindros*

La estructura lógica de un disco recibe el nombre de formato y la acción de estructurar un disco lógicamente se denomina "formatear" o dar formato. El formato de un disco depende de su tamaño y de su densidad. Los elementos lógicos de un disco son sus caras, sus pistas o cilindros, sus sectores y sus *clusters* (conjunto de sectores).

La numeración de las caras es secuencial, empezando desde cero: 0, 1, 2, 3, ..., según sea un *diskette* o un disco duro. Los *diskettes* tienen dos caras. Dependiendo de si la unidad de disco de que se dispone tiene una o dos cabezas lectoras, se podrán utilizar una o ambas caras del *diskette*, respectivamente. En los discos duros se tendrán 4, 6, 8 caras dependiendo de la capacidad del disco. La numeración de las pistas también es secuencial y va desde la pista 0 a la 39 en los *diskettes* de 5,25 pulgadas de 360 Kb, de 0 a 79 en los *diskettes* de 5,25 pulgadas de 1,2 Mb, de 0 a 305 cilindros en los discos duros de 10 Mb, etc.

A efectos del BIOS, la información se localiza en un disco por medio de un sistema de tres coordenadas formado por pista o cilindro, cara o cabeza y sector. El DOS localiza la información numerando los sectores secuencialmente. El sector 1 de la pista 0, cara 0 es el primer sector; el sector 1 de la pista 0, cara 1 es el segundo sector, y así sucesivamente.

El formato distribuye el espacio libre del disco en cuatro secciones para diferentes usos (véase figura A.5). Por orden de almacenamiento en disco están: el registro de arranque (*boot*), la tabla de localización de ficheros (FAT, *File Allocation Table*), el directorio principal o raíz y el área de datos.

A continuación se estudia en detalle cada división del disco.

A.3.1. El registro de arranque

El registro de arranque (*boot*) es un sector único que se sitúa en el sector 1, pista 0, cara 0. Este registro se graba en todos los discos al darles formato, aunque no se grabe en ellos el sistema operativo. Contiene un programa corto que activa el proceso de carga del DOS en memoria. Comprueba si el disco está formateado por el sistema y si contiene los ficheros IO.SYS o IBMBIO.COM y MSDOS.SYS o IBMDOS.COM.

Dir 1
Dir 2
Dir 3
Dir 4
Reg. arranque
FAT 1
FAT 2
FAT 3
FAT 4
Cara 0
Pista 0, cara 0

Dir 5
Dir 6
Dir 7
Datos 1
Datos 2
Datos 3
Datos 4
Datos 5
Datos 6
Cara 1
Pista 0, cara 1

Figura A.5. *Organización del espacio libre del disco*

También contiene los parámetros que forman parte del bloque de parámetros BIOS que son utilizados por el sistema para controlar las entradas/salidas por dispositivos.

Cada posición del registro de arranque contiene los siguientes parámetros:

Tabla A.2

Desplazamiento	*Longitud*	*Descripción*
3	8 bytes	Identificación del sistema.
11	1 palabra	Número de bytes por sector.
13	1 byte	Número de sectores por *cluster.*
14	1 palabra	Número de sectores reservados al principio del disco.
16	1 byte	Número de copias de la FAT.
17	1 palabra	Número de elementos del directorio raíz.
19	1 palabra	Número total de sectores en el disco.
21	1 byte	Tipo de formato.
22	1 palabra	Número de sectores por FAT.
24	1 palabra	Número de sectores por pista.
26	1 palabra	Número de caras.
28	1 palabra	Número de sectores especiales reservados.

A.3.2. La tabla de localización de ficheros (FAT)

La tabla de localización de ficheros comienza en el sector 2, pista 0, cara 0. Contiene el registro oficial del formato del disco y los mapas de la localización de los sectores utilizados por los ficheros. Al dar formato a un disco se graban dos copias de la tabla de localización, en previsión de que una se dañe y se pueda restituir por la copia. La FAT ocupa tantos sectores como sean necesarios para direccionar todos los sectores del disco.

Al principio de la tabla de localización de ficheros se almacena un registro que muestra cómo está asignado el espacio en disco. Existen dos formatos de FAT, uno de 12 bits para *diskettes* y otro de 16 bits para discos duros. Está organizada como una tabla de hasta 4096 elementos (de 0 a 4095, o de 000H a FFFH), uno para cada *cluster*. El núme-

ro que se almacena en cada elemento describe el estado y el uso del *cluster* (libre, ocupado, mal estado, etc.). En la tabla siguiente se señalan los distintos valores que puede tomar un elemento de la tabla de localización de ficheros.

Tabla A.3

Número		*Descripción*
0		*Cluster* libre y disponible para el uso.
4087	(FF7H)	*Cluster* defectuoso. No utilizable por error de formateo.
4081-4086	(FF1H-FF6H)	No se utiliza. *Cluster* defectuoso.
2-4080	(002H-FF0H)	*Cluster* utilizado por fichero.
4095	(FFFH)	Ultima parte del fichero.
4008-4094	(FF8H-FFEH)	Ultima parte del fichero. No se utiliza.

Para localizar el primer elemento de la FAT correspondiente a un fichero, se utiliza el número de *cluster* de inicio del fichero almacenado en la entrada correspondiente del directorio principal.

Cuando se borra un fichero, todos sus elementos en la tabla de localización de ficheros son marcados con el código 000H, correspondiente a un *cluster* libre y disponible para su uso.

Los *clusters* de datos están numerados desde el 2, mientras que la FAT comienza con los elementos 0 y 1. Estos dos primeros elementos de la tabla de localización de ficheros, en ambos formatos, se utilizan o están reservados para que el primer byte de la FAT pueda ser utilizado como byte de identificación, indicando el formato del disco.

A.3.3. El directorio raíz

El directorio principal o raíz ocupa siete sectores del disco. Los sectores 6, 7, 8, 9 de la pista 0, cara 0, y los sectores 1, 2, 3 de la pista 0, cara 1. Es una tabla del contenido del disco. Asigna a cada fichero del disco un elemento del directorio, que contiene cierta información, como el nombre, el tamaño, etc. Una parte de cada elemento del directorio es un puntero que indica cuál es el primer grupo de sectores utilizados por el fichero. Es

también la primera entrada del fichero en la tabla de localización de ficheros. Los subdirectorios son tratados como ficheros y, por tanto, sólo tienen asignado un elemento del directorio raíz.

Este directorio se utiliza para almacenar la información básica de los ficheros contenidos en el disco. Los datos referentes a un fichero son: nombre y extensión, tamaño, comienzo del elemento de la FAT, hora y fecha de creación o de la última modificación, y características especiales del disco.

En el directorio raíz hay un elemento por cada fichero, un elemento por cada subdirectorio y un elemento para la etiqueta del volumen de disco.

Cada elemento del directorio ocupa 32 bytes; por tanto, en un sector caben 16 elementos (un sector tiene 512 bytes; dividido entre 32 bytes que posee cada elemento, hacen 16 elementos por sector). El directorio raíz ocupa 7 sectores del disco, con un total de 112 elementos.

Los subdirectorios son tratados por el sistema operativo como ficheros que contienen una lista de nombres de ficheros y programas, por lo que el número de elementos de un subdirectorio sólo está limitado por la capacidad del disco que se esté utilizando.

Cada elemento del directorio raíz está dividido en ocho campos. En la tabla siguiente puede verse qué contiene cada campo, cuánto ocupa y cuál es su posición dentro del elemento.

Tabla A.4

Campo	*Desplazamiento*	*Descripción*	*Tamaño (bytes)*	*Formato*
1	0	Nombre del fichero.	8	ASCII.
2	8	Extensión del fichero.	3	ASCII.
3	11	Atributo.	1	Bit codificador.
4	12	Reservado.	10	No utilizado.
5	22	Hora.	2	Palabra codificada.
6	24	Fecha.	2	Palabra codificada.
7	26	Comienzo de la entrada a la FAT.	2	Palabra.
8	28	Tamaño del fichero.	4	Entero.

A continuación se estudia detalladamente cada campo de los elementos del directorio, señalando sus características más importantes.

El nombre del fichero es el primer campo de cada elemento del directorio, tiene una longitud de 8 bytes y se graba en el disco en formato ASCII (tal como aparece el nombre en pantalla). Si el nombre que se le da al fichero que se va a grabar en el disco tiene menos de 8 caracteres, el sistema operativo rellena el espacio restante con blancos por la derecha (código ASCII 32). Por ejemplo, si el nombre de un programa es VACUN.COM, el sistema grabará en el primer campo del elemento del directorio lo siguiente: VACUN___ (el carácter "_" representa un espacio en blanco). El primer byte de los ocho que tiene el campo nombre de fichero puede contener distintos códigos para indicar el estado o la situación del elemento del directorio. Si el primer byte contiene el código 00H, indica que el elemento no está utilizado por ningún fichero o subdirectorio. Si contiene el código E5H, indica que el elemento pertenece a un fichero borrado o bien que dicho elemento no ha sido utilizado nunca. Si el código es 2EH (código ASCII del punto, ".") indica que el elemento pertenece a un subdirectorio. Por último, si el código que contiene el primer byte del campo es el correspondiente a una letra, indica que el elemento está ocupado por un programa o fichero. El nombre del fichero nunca puede ser igual a 8 espacios y siempre se graba en letras mayúsculas.

El segundo campo de cada elemento del directorio raíz indica la extensión del fichero. Ocupa 3 bytes de los 32 que componen el elemento y se graba en formato ASCII. Al igual que ocurre con el nombre del fichero, si la extensión tiene menos de 3 caracteres, el sistema operativo rellena los espacios de la derecha del campo con blancos (código ASCII 32) y siempre se graba en mayúsculas. Por el contrario, la extensión de un programa o fichero sí puede quedar en blanco.

En ambos campos nunca puede haber un espacio en blanco dentro de la cadena de caracteres que componen el nombre o la extensión. No está permitido, por ejemplo, poner como nombre a un programa la siguiente cadena de caracteres: VAC 13.COM.

Cuando el elemento que se graba en el directorio raíz

pertenece a la etiqueta del volumen de disco, los dos primeros campos se tratan como uno solo de 11 bytes. En este caso está permitido introducir espacios en la etiqueta y el uso de letras minúsculas.

El tercer campo del elemento indica una característica del fichero llamada atributo. Este campo ocupa un byte y se graba en formato binario. El significado de cada bit de este byte se indica en la tabla siguiente.

Tabla A.5

Valores			*Significado*
Bit	*Decimal*	*Hexadecimal*	
00000001	1	1	Sólo lectura.
00000010	2	2	Oculto.
00000100	4	4	Sistema.
00001000	8	8	Etiqueta de volumen.
00010000	16	10	Subdirectorio.
00100000	32	20	Archivo.
01000000	64	40	No utilizado.
10000000	128	80	No utilizado.

Cada bit del campo indica un atributo distinto, pudiéndose combinar varios atributos. Un fichero que tenga como atributo 00000110 será oculto y de sistema.

El cuarto campo del elemento está reservado y no puede utilizarse. Se creó en previsión de versiones futuras del DOS, por si fuera necesario añadir información a los elementos del directorio.

La hora se señala en el quinto campo de cada elemento. Ocupa dos bytes y se graba de forma codificada. Indica la hora en la que fue creado el programa o fichero, o la hora de la última modificación. El sistema operativo usa la hora conjuntamente con el campo fecha, tratando estos dos campos como un campo entero sin signo, de 4 bytes. El campo hora por sí solo es considerado como un campo entero sin signo de 2 bytes. Para grabar la hora el sistema operativo realiza una fórmula de conversión con la cual a partir de las horas, minutos y segundos obtiene un número entero. La fórmula que utiliza es la siguiente:

Número = Hora * 2048 + Minutos * 32 + Segundos / 2

El campo fecha contiene la fecha en que fue creado el fichero o programa, o la fecha de la última modificación. Ocupa 2 bytes y se graba de forma codificada como un número entero sin signo. Al igual que con la hora, el sistema operativo utiliza una fórmula de conversión con la que a partir del año, el mes y el día obtiene un número entero. La fórmula de conversión para la fecha es la siguiente:

Número = (Año - 1980) * 512 + Mes * 32 + Día

El año mayor que se puede obtener con la fórmula de conversión es el 2108, pero el sistema operativo no admite años que soprepasen el 2099.

El séptimo campo del elemento contiene el número de *cluster* de comienzo del fichero en el área de datos del disco. Este campo actúa como puntero de entrada en la cadena de localización del fichero en la tabla de localización de ficheros. Si el elemento pertenece a un fichero sin espacio localizado en el área de datos del disco o a la etiqueta del volumen del disco, el número de *cluster* de comienzo es cero.

El último campo de cada elemento del directorio raíz indica el tamaño del fichero en bytes. Se graba como un entero sin signo, de 4 bytes. Este formato permite que un fichero pueda tener el tamaño que se quiera, pudiendo llegar hasta la capacidad total del disco.

A.3.4. El área de datos

El área de datos ocupa el resto de sectores y pistas del disco. Está organizado en *clusters* o grupos de sectores. El tamaño de un *cluster* depende del tamaño del disco y de su formato. En *diskettes* coincide el tamaño de un *cluster* con el tamaño de un sector. En discos duros, el tamaño de un *cluster* suele ser de cuatro sectores.

En el espacio de datos del disco se guardan todos los ficheros de datos, programas y subdirectorios (que actúan como ficheros de datos). Esta área del disco ocupa la parte última del mismo y es la más grande de las cuatro divisiones del disco.

El espacio en disco se va asignando a los ficheros y

programas según lo van necesitando. El espacio se asigna por *clusters* enteros. Un fichero nunca podrá ocupar, por ejemplo, un *cluster* y medio. Así, si un fichero tiene un tamaño de 720 bytes, se le asignarán dos *clusters* del espacio de datos del disco, es decir, 1024 bytes. Los ficheros se van grabando uno detrás de otro en el espacio de datos. Por ejemplo, cuando se graba un fichero que ocupa 1500 bytes, se le asignarán los tres primeros *clusters* del área de datos (1536 bytes); si a continuación se graba otro fichero de 320 bytes, ocupará el tercer sector del área de datos del disco (512 bytes). Si el primer fichero aumenta de tamaño, los nuevos datos se grabarán a partir del cuarto *cluster* del área de datos. Esto es lo que se denomina fragmentación de ficheros.

Dicha fragmentación hace que el acceso a los datos de un fichero sea más lento y consuma más recursos del sistema operativo, además de dificultar las tareas de recuperación de ficheros cuando éstos han sido borrados. Actualmente estas tareas son fáciles de realizar con ciertos programas de utilidades, como las Utilidades Norton. (Los comandos de las Utilidades Norton que deben emplearse son: DR, Recuperar Directorio, para recuperar un directorio entero y mostrar todos los ficheros que contenía; RR, Recuperación Rápida, que realiza una recuperación automática, fácil y rápida de los archivos borrados, y la opción Recuperar de la UN, Utilidad Norton principal, si se precisan procedimientos de recuperación más sotisficados.) Los ficheros fragmentados pueden reagruparse de distintas formas. Una de ellas es copiar el contenido de un disco, fichero a fichero, en otro disco. Los ficheros del disco destino estarán reagrupados y ocuparán *clusters* contiguos. Otra forma de reagrupar los ficheros es utilizando algún programa, como AD, Acelerar Disco, de las Utilidades Norton, que aumenta la velocidad de las operaciones de disco, tales como lectura/escritura, eliminando la fragmentación de los ficheros.

A.3.5. Particiones

Cada sistema operativo tiene su propia manera de dar formato y de gestionar el área de datos del disco, que resulta incompatible con la forma utilizada por otros sis-

temas operativos. Como es posible que varios sistemas operativos utilicen un mismo disco duro, se ha desarrollado una forma de dividir un disco duro en varias partes lógicas denominadas particiones. Una partición es una serie de cilindros contiguos, cuyo tamaño es especificado por el usuario y son gestionados por un único sistema operativo.

Antes de utilizar un disco duro debe ser "particionado" y hay que darle formato a cada partición con el sistema operativo que la vaya a gestionar. Generalmente, en los ordenadores personales sólo se utiliza un sistema operativo, el DOS, y por tanto el disco duro sólo contiene una partición que ocupa todo el disco. El número y tamaño de las particiones de un disco se puede variar tantas veces como se quiera, pero hay que tener en cuenta que al realizar esta operación la información contenida en cada partición se pierde, y se debe dar formato a todas las particiones con el sistema operativo correspondiente.

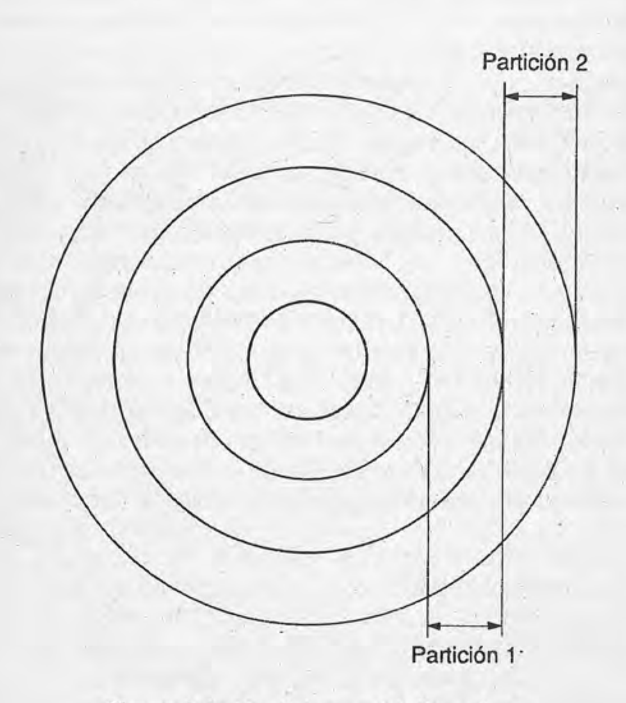


Figura A.6. *Particiones del disco duro*

Como se ha visto anteriormente, un disco utilizado por el DOS contiene en su primer sector un registro de arranque que se compone de un programa de inicialización y de cierta información acerca del disco. En un disco particionado, el primer sector (*master*) contiene un programa de inicialización y un registro que indica cómo está particionado el disco. Este registro señala cuántas particiones hay en el disco, además del tamaño de cada una, su localización dentro del disco y qué partición está activa, es decir, qué sistema operativo arrancará cuando encendamos el ordenador. Por ejemplo, si se tiene dos particiones en él disco, una gestionada por el sistema operativo DOS y otra por el sistema operativo XENIX, y la partición que está activa es la del DOS, al encender el ordenador se empezará trabajando en DOS. Para trabajar con el sistema operativo XENIX, se tiene que cambiar la partición activa del disco (con el comando FDISK del DOS).

El programa de inicialización del *master* es un programa corto que cede el control al programa de arranque de la partición activa.

A.4. Programas ejecutables .COM y .EXE

En el entorno de programación del DOS hay dos tipos de programas ejecutables, los programas con extensión .COM y .EXE.

Después de escribir un programa en cualquier lenguaje de programación de alto nivel, se debe compilar para obtener un código ejecutable. La mayoría de los compiladores que hay en el mercado producen un programa ejecutable con extensión .EXE. Para obtener un programa con extensión .COM se tiene que utilizar el comando EXE2BIN del sistema operativo DOS.

Existen ciertas diferencias entre los dos tipos de programas ejecutables. Los programas con extensión .COM son más compactos y rápidos que los de extensión .EXE, puesto que no contienen información de reubicaciones y sólo pueden tener una longitud máxima de 65536 bytes (el tamaño de un segmento de memoria), menos la longitud del prefijo de segmento de programa (PSP), que son 256 bytes y una palabra (dos bytes) reservada para la pila

de datos. El DOS no comprueba si los programas .COM contienen código ejecutable, como lo hace con los programas con extensión .EXE. Al ejecutar un programa .COM el sistema operativo lo carga en memoria y salta directamente a la dirección de entrada o comienzo del programa, que está justamente después del PSP, sin hacer ninguna comprobación. Los programas con extensión .COM se cargan siempre en el desplazamiento 0100H del segmento de memoria que les asigna el sistema operativo, debido a que el PSP ocupa 0FFH bytes (256 bytes). Al ejecutar un programa .COM todos los registros de segmento de memoria apuntan al mismo segmento, que es el segmento que ocupa el código del programa. Los programas .COM terminan siempre con la función del sistema operativo 4CH de la interrupción 21H (esta función devuelve un código de control que puede ser tratado con la instrucción IF ERRORLEVEL de los ficheros *batch*) o con la interrupción 20H. En la versión 1.0 del MS-DOS, los programas con extensión .COM terminan con la función 00H de la interrupción 21H.

El tamaño de los programas con extensión .EXE sólo está limitado por el tamaño de la memoria del ordenador. Al ejecutar un programa de este tipo, el código del programa, los datos y la pila se sitúan en segmentos distintos. El código del programa se sitúa a continuación del PSP. Los distintos segmentos de memoria que ocupa un programa .EXE pueden estar en cualquier orden; por ejemplo, primero el segmento de código, después el segmento de datos y, por último, el segmento de pila; o bien situarse en el orden segmento de datos, segmento de pila y segmento de código de programa.

Los programas con extensión .EXE disponen de un encabezamiento, o bloque de información de control, que posee un formato característico. El tamaño de éste varía según el número de instrucciones que deban ser reubicadas durante la carga del programa. El tamaño del bloque de información de control deberá ser siempre múltiplo de 512 bytes. Antes de ceder el control al programa .EXE que se va a ejecutar, el DOS comprueba que éste contenga código ejecutable y se encarga de calcular los valores iniciales de los registros de segmento de memoria. Estos programas siempre terminan con la función del sistema operativo 4CH de la interrupción 21H.

A.5. Programas transitorios y programas residentes

Un programa es un conjunto de instrucciones que se ejecutan secuencialmente, reciben una información de entrada, la procesan y devuelven una información de salida.

Al ejecutar un programa el sistema operativo realiza los siguientes pasos:

1. Lee el código del disco y lo lleva a la memoria del ordenador, más concretamente a la zona de programas transitorios, mediante la función o servicio EXEC.
2. Le cede el control a la CPU.
3. Cuando el programa termina, lo borra de la memoria.

Los programas que actúan de esta forma se denominan programas transitorios (véase figura A.7).

Por el contrario, los programas que una vez que terminan de ejecutarse no son borrados de la memoria de programas transitorios (TPA), se denominan programas residentes en memoria (véase figura A.8). Estos programas pueden ser llamados de diversas formas y por diversas causas: al pulsar una tecla, al pasar cierto tiempo, al solicitar un servicio del sistema operativo, etc.

Como ya se ha indicado, los programas .COM y .EXE terminan con instrucciones que invocan servicios del sistema operativo que borran el programa de la memoria al acabar de ejecutarse. Existen algunos servicios del sistema operativo que hacen que se termine la ejecución de un programa, pero no lo borran de la memoria. Estos servicios o funciones son: la interrupción 27H y la función 31H de la interrupción 21H.

A.6. Servicios del DOS o interrupciones

Se denomina interrupción a una señal que hace que la unidad central de proceso del ordenador suspenda la tarea que está realizando y transfiera el control a un programa especial llamado gestor de interrupciones. El gestor de interrupciones se encarga de determinar la causa

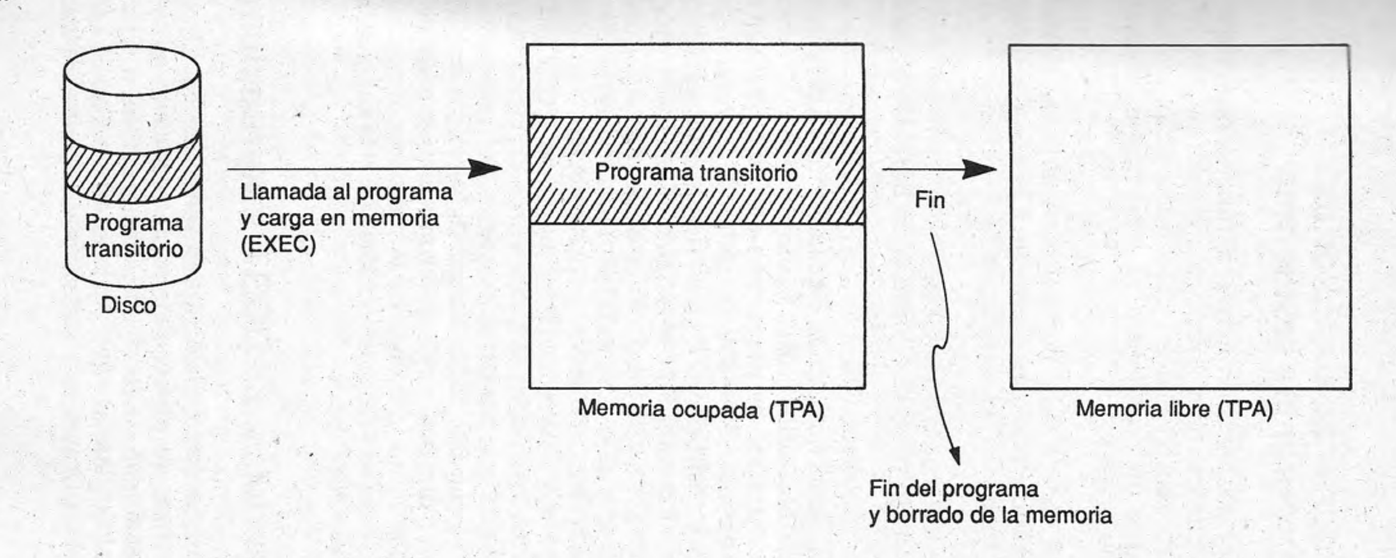


Figura A.7. *Programa transitorio*

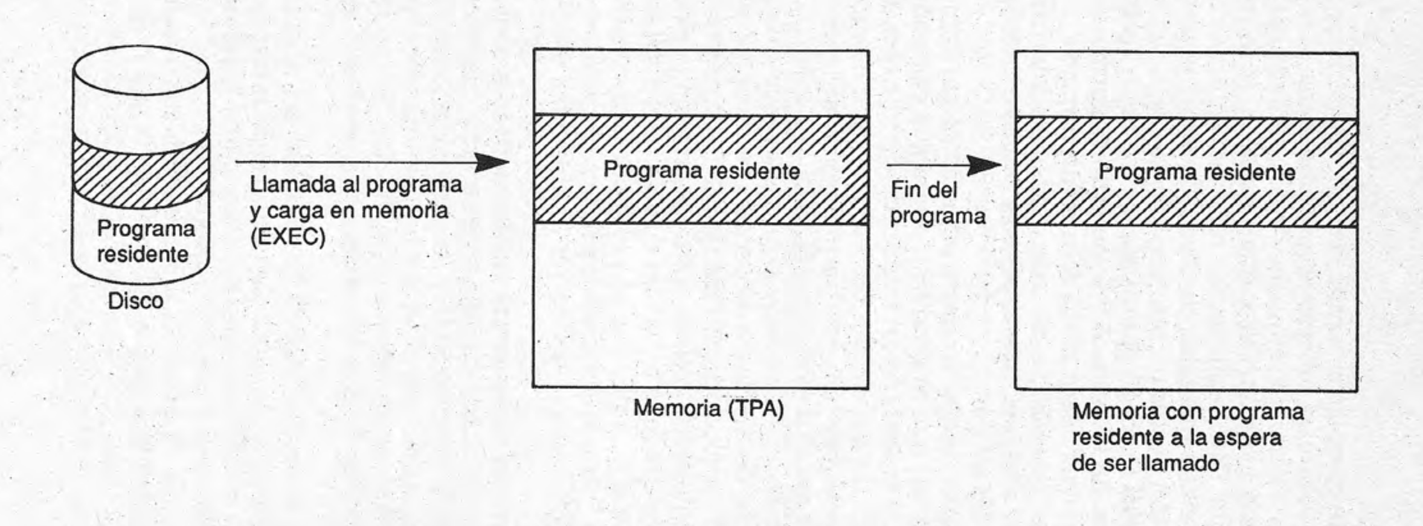


Figura A.8. *Programa residente*

de la interrupción, llevar a cabo las acciones pertinentes y devolver el control al proceso original que había sido suspendido. Este gestor se compone de una serie de rutinas, una para cada interrupción, y está situado en la memoria del ordenador. Para localizar cada rutina en la memoria del ordenador, existe una tabla de direcciones que apuntan a las rutinas del gestor de interrupciones. Cada elemento de la tabla está formado por un número llamado vector de interrupción. En la figura A.9 se muestra de forma esquemática el proceso de ejecución de una interrupción.

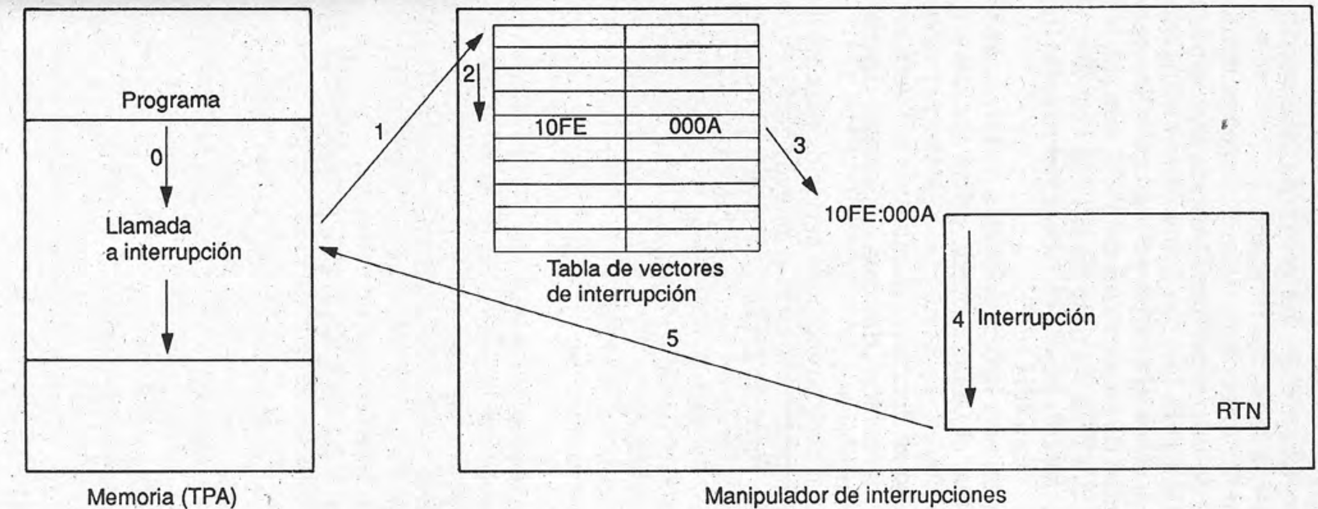
Hay tres tipos básicos de interrupciones: interrupciones *hardware* internas, interrupciones *hardware* externas e interrupciones *software*.

Las interrupciones *hardware* internas se generan por acontecimientos que se producen durante la ejecución de programas, por ejemplo un intento de dividir una cantidad por cero. La asignación de este tipo de sucesos a un vector de interrupción está implementada dentro del procesador y no es posible modificarla.

Las interrupciones *hardware* externas son activadas por controladores de dispositivos periféricos o coprocesadores (como el coprocesador aritmético 8087). Se generan por acontecimientos catastróficos, como un error en la paridad de memoria o un fallo en la alimentación del dispositivo. Este tipo de interrupciones se gestionan a través de un dispositivo llamado controlador de interrupciones programable (PIC) y no pueden ser modificadas mediante programas.

Las interrupciones *software* pueden activarse de manera sincronizada desde cualquier programa, mediante la ejecución de la instrucción INT. Las interrupciones 20H a 3FH son empleadas por el DOS para comunicarse con sus módulos y con los programas de aplicación. Por ejemplo, al administrador de funciones del DOS se accede ejecutando INT 21H.

Los 1024 bytes iniciales de la memoria del ordenador reciben el nombre de tabla de vectores de interrupción (TVI). Cada posición de esta tabla tiene una longitud de 4 bytes y pertenece a una interrupción. La TVI puede contener hasta 256 vectores de interrupción (resultado de dividir 1024 bytes entre los 4 bytes que ocupa cada elemento de la tabla). Un elemento de la tabla de vectores de interrupción contiene el número de segmento de la



0. Ejecución de un programa.
1. Llamada al manipulador de interrupciones.
2. Búsqueda del vector de interrupción.
3. Llamada a la interrupción.
4. Ejecución de la interrupción.
5. Retorno al programa.

Figura A.9. *Ejecución de una interrupción*

memoria y el desplazamiento del gestor de interrupciones correspondiente a cada interrupción.

Las primeras posiciones de la tabla de vectores apuntan a los gestores de interrupciones *hardware*, que son las interrupciones 01H a 1FH. Los elementos intermedios son utilizados por el sistema operativo para guardar los vectores de interrupción de los servicios del DOS, que son las interrupciones 20H a 3FH. El resto de la tabla está disponible para que el usuario pueda construir sus propias interrupciones, que serían 40H a FFH.

Antes de crear nuevas interrupciones e incluir nuevos vectores en la tabla, el usuario puede optar por desviar los vectores ya existentes hacia rutinas o gestores creados por él. Este proceso se denomina cambio de los vectores de interrupción, "parcheado" de interrupciones o desplazamiento de interrupciones.

El DOS provee al programador de servicios para realizar estos cambios y permitirle instalar nuevos gestores de interrupciones. La función 25H de la interrupción 21H permite al usuario asignar un vector de interrupción de la tabla de vectores a un gestor cualquiera. La función 35H de la interrupción 21H permite al programador obtener el vector de interrupción de la TVI de la interrupción que indique. Por último, la función 31H de la interrupción 21H permite al programador crear programas residentes en memoria.

Estas tres funciones hacen que un usuario experto pueda examinar o modificar el contenido de la tabla de vectores de interrupción del sistema y reservar memoria para el uso del nuevo gestor de interrupciones, sin necesidad de interferir con otros procesos del sistema o producir conflictos en la utilización de memoria.

En el apéndice E se encuentra una lista detallada de cada uno de los servicios de interrupción que permite este sistema operativo.

Apéndice B

El Viernes-13 a fondo

B.1. Introducción

El Viernes-13 es fundamentalmente un virus, pero tiene características o propiedades de los caballos de Troya y de las bombas lógicas. Es un virus porque es capaz de realizar copias de sí mismo adhiriéndose a los ficheros ejecutables .EXE y .COM, consiguiendo una ejecución parasitaria, es decir, sin que el programador lo llame explícitamente. Tiene características del caballo de Troya, o mejor dicho, convierte los programas que contamina en caballos de Troya. Es decir, siguen teniendo su apariencia normal pero llevan en su seno una serie de instrucciones añadidas que sólo se ejecutan una vez al poner en marcha el programa portador. Por último, posee una función de bomba lógica porque ante un determinado hecho se activa y realiza acciones inesperadas para el usuario.

B.2. Versiones del Viernes-13

Existen varias versiones del Viernes-13, también llamado virus de Jerusalén, PLO (en castellano OLP, Organización para la Liberación de Palestina) o virus de Israel. La primera versión contenía un presunto fallo de programación, que consistía en contaminar los ficheros .EXE tantas veces como fueran ejecutados hasta que en una ejecución se desbordaba la memoria. Para remediar este fallo y/o ampliar las acciones perniciosas del Viernes-13 surgieron las distintas versiones que se relacionan a continuación.

Jerusalén-B

Esta versión detecta si un fichero .EXE ya está contaminado o no, infectándolo una sola vez. Por lo demás, su forma de actuar es igual a la del virus original.

Jerusalén-C

Para no revelar su presencia en la memoria, no retrasa la velocidad de proceso del ordenador. Esta versión también se llama Nuevo Jerusalén.

Jerusalén-D

Esta versión sustituye el borrado de los ficheros que se ejecutan en viernes 13 por el borrado de las dos FAT que contiene el disco. Esta acción sólo la realiza a partir de 1990.

Jerusalén-E

Actúa exactamente igual a la versión original, con la salvedad de que se activa solamente a partir de 1992.

Century

También llamado virus de Oregón. Es similar al virus de Jesuralén-C. No se activa hasta el 1 de enero del año 2000. Su acción destructiva se centra en el borrado de las FAT de todos los discos conectados al sistema. A continuación escribe ceros aleatoriamente en algún sector de cualquiera de los discos conectados.

Century-B

Es similar al Century, con la excepción de que espera a que se ejecute el comando externo del sistema operativo BACKUP.COM y entonces rellena con basura los ficheros que se guarden en la copia de seguridad que realiza este programa.

B.3. Actuación del virus

El Viernes-13 está escrito en lenguaje ensamblador de los procesadores INTEL 8088/8086. Debido a que su programador utilizó solamente funciones e instrucciones estándares de lenguaje, el virus de Jerusalén puede funcionar perfectamente en los sistemas operativos MS-DOS, Concurrent DOS y DR-DOS.

La forma de actuar del Viernes-13 se puede dividir en cuatro fases: carga, infección, destrucción y acciones específicas.

La carga del virus en memoria se realiza cuando se ejecuta un programa infectado, ya sea un programa con extensión .COM o con extensión .EXE. El Viernes-13, al cargarse en memoria, realiza una serie de acciones para configurar su entorno de actuación y a continuación se hace residente en memoria a la espera de que se den las condiciones necesarias para realizar sus acciones perniciosas. La primera acción que realiza el virus es comprobar si ya está residente en memoria. Esto lo hace creando un nuevo servicio del sistema operativo, función E0H de la interrupción 21H. Al llamar a esta función, si el programa está residente en memoria la interrupción devuelve el valor 03H en el registro AH del procesador. Si no está residente devuelve el mismo valor de llamada a la función, es decir, E0H en el registro AH.

El código en lenguaje ensamblador que define esta acción es el siguiente:

```
001        CLD
002        MOV     AH,E0H
003        INT     21H
004        CMP     AH,E0H
005        JNB     etiq1
006        CMP     AH,03H
007        JB      etiq1
```

En las líneas 002 y 003 mueve el valor E0H al registro AH y llama a la interrupción 21H. En la instrucción de la línea 004 comprueba si el programa no está residente, comparando el valor devuelto en el registro AH con el valor E0H, que indica que el programa no está residente. En la línea 005 salta a la instrucción situada en la etiqueta "etiq1" si el contenido del registro AH es mayor o igual que el valor E0H, es decir, si el programa no está residente en memoria. Por último, en las líneas 006 y 007 comprueba si el valor devuelto en el registro AH es 03H, y salta a la instrucción de la etiqueta "etiq1" si el valor contenido en el registro AH es menor que el valor 03H.

Como se ha dicho anteriormente, antes de hacerse residente el virus realiza una serie de acciones. Primero defi-

ne tres nuevas funciones o servicios de la interrupción 21H del sistema operativo; éstas son las funciones DDH y DEH, junto con la función E0H anteriormente descrita.

A continuación desplaza tres interrupciones del sistema operativo. La interrupción 21H, para tomar el control de los servicios del DOS, especialmente el servicio 4BH o función EXEC (ejecutar un programa) para poder infectar los programas antes de que sean ejecutados. La interrupción 24H, para controlar los errores que se puedan producir mientras el virus está realizando alguna acción, y la interrupción 8H, para controlar el reloj del sistema. La interrupción 24H es desplazada para ocultar el virus y sus acciones. Si por alguna razón se produce un error en alguna acción del virus, éste evita que el mensaje correspondiente a ese error salga en la pantalla del ordenador y así puede seguir residente en memoria a la espera de las condiciones adecuadas para realizar sus acciones de contagio o destrucción (como se comentará más adelante esta medida del virus falla cuando el disco está protegido físicamente). A continuación se muestra cómo el virus desplaza la interrupción 21H. El desplazamiento se realiza cambiando el vector que apunta al manipulador de la interrupción por uno nuevo. El código en lenguaje ensamblador para realizar esta operación es el siguiente:

```
001       MOV     AH,35H
002       MOV     AL,21H
003       INT     21H
004       MOV     CS:[0017H],BX
005       MOV     CS:[0019H],ES
006       PUSH    CS
007       POP     DS
008       MOV     DX,025BH
009       MOV     AH,25H
010       MOV     AL,21H
```

En las intrucciones de las líneas 001, 002 y 003, llama a la función 35H, valor que se lleva al registro AH, de la interrupción 21H, valor que se lleva al registro AL. Esta función obtiene el vector de interrupción de la interrupción especificada en el registro AL y devuelve el vector en los registros BX y ES. En el registro ES devuelve el segmento de memoria donde está ubicado el gestor de la in-

terrupción y en el registro BX devuelve el desplazamiento (*offset*) dentro del segmento. A continuación se guardan el desplazamiento y el segmento del antiguo vector en las direcciones de memoria CS:[0017H] y CS:[0019H], respectivamente, en las líneas 004 y 005. Por último, en las líneas 006, 007, 008, 009 y 010 se introduce el nuevo vector en la tabla de vectores de interrupción. Mediante la función 25H, valor que se lleva al registro AH, de la interrupción 21H, valor que se lleva al registro AL, se introduce el nuevo vector cuyo segmento debe estar introducido en el registro DS y su desplazamiento en el registro DX. En este caso el nuevo vector de la interrupción 21H será CS:025BH.

La última acción que realiza, antes de hacerse residente, es comprobar la fecha del sistema. Primero captura la fecha mediante la función 2AH de la interrupción 21H y a continuación comprueba si el año es 1987, si el día de la semana es viernes y si el día del mes es 13. El código correspondiente a esta acción es el siguiente:

```
001             PUSH   DS
002             PUSH   ES
003             PUSH   AX
004             PUSH   BX
005             PUSH   CX
006             PUSH   DX
007             MOV    AH,2AH
008             INT    21H
009    CFECH:   MOV    BYTE PTR CS:[000EH],00H
010             CMP    CX,07C3h
011             JE     PTERM
012             CMP    AL,05H
013             JNE    CINT8
014             CMP    DL,0DH
015             JNE    CINT8
016             INC    BYTE PTR CS:[000EH]
017             JMP    SHORT PTERM
018             NOP
```

En las líneas 001 a 006 introduce los valores contenidos en los registros del procesador DS, ES, AX, BX, CX y DX en la pila de datos. A continuación, en las líneas 007 y 008 lleva al registro AH el valor 2AH, que es la función de la

interrupción 21 que busca la fecha del sistema operativo, y llama a la interrupción 21H. Esta función del sistema devuelve el año en el registro CX, el día de la semana en el registro AL y el día del mes en el registro DL. En la instrucción 009 se pone a cero la posición de memoria 000EH del segmento de código de programa, que posteriormente será utilizada para comprobar si la fecha es viernes 13. En la instrucción de la línea 010 comprueba si el año es 1987 (07C3H en hexadecimal). Si el contenido del registro CX es igual a 07C3H (1987), en la línea 011 salta a la instrucción de la etiqueta PTERM, donde finaliza el programa y se hace residente en memoria. Por tanto, si el año es 1987, aunque estemos en un viernes 13 el virus no desarrollará ninguna acción destructiva, solamente realizará acciones de infección. Si el año no es 1987, se ejecuta la instrucción de la línea 012, donde el virus comprueba si el día de la semana es viernes (05H o 5 en decimal). Si el contenido del registro AL no es igual a 05H, en la línea 012, se transfiere el control a la instrucción de la etiqueta CINT8. Si es igual se ejecuta la instrucción de la línea 014, donde se comprueba si el día del mes es 13 (0DH en hexadecimal). Si el contenido del registro DL no es igual a 0DH, en la línea 14, se salta a la etiqueta CINT8; en caso contrario, es decir, si la fecha es viernes 13 de un año que no sea 1987, se ejecutará la instrucción de la línea 016, y se incrementará en uno la posición de memoria 000EH del segmento de código de programa para indicar que la fecha del sistema es viernes 13. A continuación se salta a la etiqueta PTERM, donde termina el programa y se hace residente.

Si el Viernes-13 comprueba que no está residente en memoria(función E0H de la interrupción 21H), después de realizar las acciones que se han descrito, se hace residente en memoria mediante la función 31H de la interrupción 21H. Esta acción la realiza mediante las siguientes instrucciones:

```
MOV     AH,31H
MOV     DX,0600H
MOV     CL,04H
SHR     DX,CL
ADD     DX,+10H
INT     21H
```

En la instrucción de la línea 001 lleva al registro AH el valor 31H, que es la función de la interrupción 21H que permite terminar un programa y que éste continúe residente en memoria. En las líneas 002 a 005 lleva a DX el valor 70H, que es el número de párrafos de memoria que se reservarán para el programa residente, en este caso el Viernes-13. Por último, en la línea 006 llama a la interrupción 21H para hacerse residente.

Una vez que el virus está residente en memoria, queda a la espera de que se ejecuten programas para infectarlos. Cada vez que se ejecuta un programa .COM o .EXE, el sistema operativo llama a la función EXEC (4BH) de la interrupción 21H, función que el virus ha redefinido. Antes de que se ejecute el programa, el virus lleva a cabo una serie de acciones. Si el programa que se va a ejecutar tiene extensión .COM, comprobará si ya está infectado. Si no lo está, el virus se adherirá al programa aumentando su longitud en 1813 bytes. Si el programa ya está infectado, pasa el control al mismo para que se ejecute con normalidad. Al ejecutar un programa con extensión .EXE, el virus no realiza ninguna comprobación de infección, y así este tipo de programas se infectan tantas veces como sean ejecutados hasta que se produzca un desbordamiento de la memoria. La primera vez que un programa con extensión .EXE es infectado aumenta su tamaño en 1813 bytes; las infecciones posteriores lo hacen crecer en 1808 bytes. Una vez infectado el programa, bien tenga extensión .COM o .EXE, el virus lo graba en el disco y le pasa el control para que se ejecute con normalidad. En los programas con extensión .COM el virus se adhiere al principio del programa junto con la cadena de caracteres "sUMsDos". En los programas con extensión .EXE lo hace al final, tantas veces como se ejecute el programa, teniendo que reajustar la cabecera de éste cada vez que lo infecta.

El virus ha de tomar una serie de precauciones a la hora de infectar un programa para evitar ser descubierto. Como primera medida evita contaminar el programa COMMAND.COM, programa que utilizan otros virus para su propagación y, por tanto, muy controlado por los usuarios.

Si el programa tiene atributo de sólo lectura y, por consiguiente, está protegido contra escritura, el Viernes-13 cambia el atributo del fichero a lectura/escritura y una

vez infectado y grabado en disco vuelve a cambiar el atributo a sólo lectura, evitando así ser descubierto por un error de escritura. Por esta razón, el cambio de este atributo en los programas con extensiones .COM o .EXE no sirve como protección contra este virus, aunque sí pueda servir para otros. El manejo de los atributos de los programas lo realiza el virus mediante tres acciones: buscar y guardar los atributos del fichero, poner los nuevos atributos y restaurar los antiguos.

Estas tres acciones del virus se explican más detenidamente a continuación:

- Buscar y guardar atributos.

```
001        MOV    AX,4300H
002        INT    21H
003        MOV    CS:[0072H],CX
```

La función 43H, valor que se lleva al registro AH, de la interrupción 21H, sirve para el manejo de los atributos del programa. Al llevar el valor 00H al registro AL se indica a la interrupción que la acción que se va a realizar es la obtención de los atributos del programa (líneas 001 y 002). Este servicio de la interrupción 21H devuelve en el registro CX los atributos del programa que se graban en la posición de memoria CS:[0072H] en la línea 003.

- Poner nuevos atributos.

```
001        XOR    CX,CX
002        MOV    AX,4301H
003        INT    21H
```

En la línea 002 se lleva al registro AL el valor 01H, que indica a la función 43H, de la interrupción 21H, que se van a modificar los atributos del programa. Para realizarlo se han de introducir en el registro CX. En este caso se ponen todos los valores iguales a cero en la línea 001, con lo que el fichero queda definido como de lectura/escritura. Por último, en la línea 003 se llama a la interrupción 21H para que se ejecute la función especificada.

- Restaurar antiguos atributos.

```
001        MOV   CX,CS:[0072H]
002        MOV   AX,4301H
003        INT   21H
```

En la línea 001, recupera los atributos del programa que se guardaron en la posición de memoria CS:[0072H] y los lleva al registro CX. En las líneas 002 y 003 llama a la función 43H, valor que se lleva al registro AH, de la interrupción 21H, indicando que se van a grabar los atributos llevando el valor 01H al registro AL.

Cada vez que el sistema operativo reescribe un fichero o programa en el disco, actualiza la fecha y la hora para indicar cuándo se modificó por última vez el programa. Es normal que en muchas empresas se lleve un control de dichos parámetros para conocer el desarrollo de una aplicación o para desarrollar la documentación de un proyecto. El Viernes-13, para evitar ser descubierto por este motivo, obtiene y guarda la fecha y la hora del programa que va a infectar antes de modificarlo y las restaura una vez grabado el programa en el disco.

Esta doble acción se analiza detalladamente a continuación:

- Buscar y guardar la fecha y la hora.

```
001        MOV   AX,5700
002        INT   21H
003        MOV   DS:[0074H],DX
004        MOV   DS:[0076H],CX
```

En las líneas 001 y 002 llama a la función 57H, valor que se lleva al registro AH, de la interrupción 21H. Se indica con ello que se va a leer la fecha y la hora, llevando el valor 00H al registro AL. Este servicio del sistema operativo devuelve la fecha en el registro DX y la hora en el registro CX. En las líneas 003 y 004 guarda la fecha y la hora en las posiciones de memoria del segmento de datos DS:[0074H] y DS:[0076H], respectivamente.

- Restaurar la fecha y la hora.

```
001        MOV    DX,DS:[0074H]
002        MOV    CX,DS:[0076H]
003        MOV    AX,5701H
004        INT    21H
```

Primero recupera la fecha y la hora guardadas en las posiciones de memoria DS:[0074H] y DS:[0076H] y las lleva a los registros DX y CX, en las líneas 001 y 002, respectivamente. A continuación, en las líneas 003 y 004 llama a la función 57H, valor que se lleva al registro AH, de la interrupción 21H. Se indica con ello que se va a grabar la fecha y la hora, llevando el valor 01H al registro AL.

Cuando se intenta ejecutar un programa, el sistema operativo lo busca por el directorio de trabajo y por los directorios especificados en el comando PATH. Si no encuentra el programa, el sistema emite un mensaje de error indicando este hecho. El Viernes-13 hace lo mismo para encubrirse, comprobando si el nombre del programa que se llama es correcto y existe en los directorios anteriormente citados.

Por último, el virus comprueba si tiene suficiente espacio en el disco para grabar el programa infectado. Si no tiene suficiente espacio, no infecta el programa para evitar delatarse.

La acción destructiva del Viernes-13 necesita, primero, que la fecha del ordenador sea viernes 13 de un año distinto a 1987 y, segundo, que el virus esté residente en memoria. Si se cumplen ambas condiciones, cualquier programa que se intente ejecutar, ya tenga extensión .COM o .EXE y ya esté contaminado o no, es borrado automáticamente sin permitírsele la ejecución. De este modo de actuar se pueden sacar varias conclusiones. Primera, el programa que lleva el virus a la memoria por primera vez, no se borra por no cumplirse la segunda condición de las dos necesarias para que se produzca la acción destructiva del virus. Segunda, la acción destructiva no reconoce si el programa que se manda ejecutar está contaminado. El virus puede destruir un fichero contaminado, no se respeta a sí mismo, pudiendo llegar a autodes-

truirse. Tercera, el programa que llevó el virus a la memoria puede borrarse si se ejecuta por segunda vez.

La acción destructiva del virus se realiza previamente a la ejecución de cualquier programa. Al intentar ejecutar un programa y siempre que se cumplan las dos condiciones descritas anteriormente, el Viernes-13 borrará el fichero. El sistema operativo al ir a ejecutar el programa responderá con el mensaje de error de comando o nombre de fichero incorrecto, puesto que éste ya no existe.

Los programas borrados por el Viernes-13 pueden ser recuperados mediante programas de utilidades, como las Utilidades Norton (comando RR, recuperación rápida de ficheros). La recuperación se puede conseguir porque el borrado se realiza mediante la función 41H de la interrupción 21H, la misma que utilizan los comandos internos DEL y ERASE del sistema operativo. Esta función graba el código E5H en el primer carácter del nombre del fichero, indicándole al sistema operativo que el fichero está borrado, pero manteniendo los datos o el código del programa en el área de datos del disco.

Si el virus intenta borrar un programa situado en un *diskette* con protección física, se descubre con ayuda de la solapa adhesiva que cubre la muesca del borde derecho del *diskette*. Al ir a borrar el programa, el Viernes-13 deja por un momento el control al sistema operativo y éste, al encontrar la protección física del disco, devuelve un mensaje de error indicándolo. Esto delata al virus, ya que para ejecutar un programa, el sistema operativo sólo tiene que leer el disco y nunca escribir en él.

Las acciones propias del Viernes-13 son aquellas que no se pueden catalogar dentro de las acciones anteriores. Su misión es perjudicar el buen funcionamiento del ordenador. El virus realiza dos acciones específicas: ralentizar la velocidad de proceso del ordenador y deformar lo que aparece en la pantalla.

Si el virus detecta que no es viernes 13, activa una rutina a la que se accede por la interrupción 8H del sistema operativo. Esta interrupción la genera la unidad central de proceso 18,2 veces por segundo y le sirve al ordenador para tener noción del tiempo. A la media hora de tener el virus en memoria, los procesos realizados por el ordenador se hacen más lentos hasta el punto de tener que apagar el ordenador y volver a arrancar. La ralentización

puede llegar hasta un 50 por 100 de la velocidad original.

También a la media hora de tener el virus residente en memoria, el Viernes 13 muestra en la parte superior izquierda de la pantalla una ventana, desplazándola hacia arriba. Esta acción la realiza mediante la interrupción 10H de la ROM BIOS. El código en lenguaje ensamblador se detalla a continuación:

```
001    MOV    AX,0602H
002    MOV    BH,87H
003    MOV    CX,0505H
004    MOV    DX,1010H
005    INT    10H
```

En la línea 001 lleva al registro AH el valor 06H, que es la función de la interrupción 10H que inicializa o desplaza una ventana de la pantalla hacia arriba. Al llevar el valor 02H al registro AL, está indicando a la interrupción que la ventana se va desplazar dos líneas hacia arriba. Si hubiese movido el valor 00H al registro AL, en vez de mover la ventana se habría inicializado con espacios en blanco. En la línea 002 lleva a BH el atributo que se va a usar en el área o ventana creada, en este caso el 87H. A continuación, en las líneas 003 y 004 define las dimensiones de la ventana que se va a desplazar. Lleva al registro CL la fila (coordenada x) y al registro CH la columna (coordenada y), de la esquina superior izquierda de la ventana. A los registros DL y DH lleva la fila y la columna, respectivamente, de la esquina inferior derecha de la ventana. Por tanto, la ventana tiene unas dimensiones de 5 por 5 caracteres. Por último, en la línea 005 llama a la interrupción 10H, que produce el desplazamiento de la ventana definida.

Por este motivo los rusos han llamado al Viernes-13 el virus Black Hole (agujero negro), ya que la ventana creada en la parte superior izquierda de la pantalla es de color negro.

En resumen, se puede afirmar que el Viernes-13 es un virus bastante elaborado y sutil en su forma de actuar, lo que le ha permitido extenderse por todo el mundo y ser el virus más conocido por los usuarios de ordenadores personales. A pesar de su sutileza contiene algunos fallos que permiten detectarlo, siempre y cuando se lleve un

perfecto control del ordenador y de los discos que se utilicen. Su medio de difusión son los programas con extensiones .COM y .EXE. Por ello se aconseja que si se recibe algún programa de procedencia dudosa, se tenga en cuarentena y se realicen las operaciones preventivas vistas en el capítulo 4 de esta guía.

Si se conoce la existencia del virus en el ordenador, es muy difícil que pueda llegar a borrar programas y datos. Por eso, si se actúa con tranquilidad y prudencia, no se tendrá ningún problema para eliminar el Viernes-13.

Apéndice C

Relación de virus conocidos

Nombre: **Alameda**
Tipo: Sector de arranque
Sistema afectado: MS-DOS

Historia

Se cree que se desarrolló en el Merritt College, California (Estados Unidos), a principios del año 1988.

Descripción

Infecta el sector de arranque de los *diskettes* de 5,25 pulgadas. El virus Alameda crea un nuevo sector de arranque y no lo protege contra escritura, provocando que éste pueda ser reescrito. Si esto ocurriera, se perderían todos los datos del *diskette*.

Graba el sector de arranque original en la pista 39, sector 8, cara 0, del *diskette*. Tiene un contador para almacenar el número de veces que infecta a otros *diskettes*, pero no se usa para activar el virus.

Este virus no funciona si el ordenador posee un microprocesador de la serie 286.

Versiones

Alameda-B

Esta versión funciona con procesadores de la serie 286.

Alameda-C

Esta versión utiliza el contador y hace que el sector de arranque de la copia número 100 se desactive y no se pueda utilizar el *diskette*.

Nombre: **Aldus (MACMAG, Drew)**
Tipo: Sector de arranque
Sistema afectado: Macintosh

Historia

Este virus lo escribió un periodista canadiense para la revista *MAC magazine*. También apareció en dos BBS, *Compuserv* de Columbus (Ohio) y *Gene* de Rockville, en Estados Unidos.

Descripción

El 2 de marzo de 1988 el virus se activaba sacando un mensaje de paz por pantalla, a continuación el virus se autoborraba.

Versiones

Peace

Es una modificación de Aldus manipulada por grupos pacifistas. El virus se propaga por medio de la inserción de un servicio INIT 6, cuyo nombre es RR, en el fichero de sistema. No infecta programas de aplicación y se propaga sólo en ficheros de sistema presentes en discos duros o *diskettes*.

Nombre: **Anti**
Tipo: Programa
Sistema afectado: Macintosh

Historia

Apareció inicialmente en Francia. También se encontró en París, Marsella y en muchas otras ciudades. Thierry Lalettre, moderador jefe del Macintosh *forum* de Calva-Com, alertado por varios usuarios envió copias del virus a muchos programadores de vacunas y detectores para Macintosh.

Descripción

El virus Anti se adhiere al final de los programas de aplicación. Cuando se llama la aplicación, cambia el código del programa infectado para ejecutarse en primer lugar. Al ejecutar una aplicación infectada, el virus infecta la parte del sistema que está residente en memoria pero no infecta los ficheros del sistema.

Este virus puede detectarse de varias maneras:

1. Añade 1344 bytes al código del programa infectado. La fecha y la hora del programa se cambian por la fecha y la hora en que se produce la infección.
2. Contiene siete veces la cadena "$16252553". La última aparición de esta cadena está situada 43 bytes antes del final del programa infectado. El virus utiliza esta cadena para detectar si ya se ha infectado un sistema o una aplicación.
3. También contiene una cadena de 9 caracteres, seguida por la cadena "#000000" y la palabra "Anti", de ahí su nombre.

Versiones

No tiene versiones conocidas.

Nombre: **ARPANET DATA**
Tipo: Gusano
Sistema afectado: Red ARPANET

Historia

Se descubrió el 27 de octubre de 1980 en la red ARPANET (*Advanced Research Projects Administration Network*). Se cree que la infección se originó en Los Angeles. El gusano se expandió a través de la red infectando todos los nodos hasta que produjo un colapso. La red permaneció tres días fuera de servicio hasta que se restablecieron las comunicaciones.

Descripción

El gusano se expandió utilizando el sistema de mensajes de la red, camuflado como un mensaje de estado. De esta manera contaminó el programa de gestión de mensajes, impidiendo la salida de mensajes de estado y autorizando sólo la salida de mensajes contaminantes. Como consecuencia, la red se saturó y el sistema se vino abajo.

Versiones

No tiene versiones conocidas.

Nombre: **Brain**
Tipo: Sector de arranque
Sistema afectado: MS-DOS

Historia

Fue detectado por primera vez en el *Journal-Bulletin* de Providence, Rhode Island (Estados Unidos). Sus creadores fueron Basit y Alvi Amjad, de Lahore, Pakistán. Su idea original era hacer un seguimiento de las copias piratas de los programas que llevaban su *copyright*.

Descripción

El virus infecta los *diskettes* que contienen el sistema operativo, es decir, aquellos *diskettes* con los que se puede arrancar el ordenador, instalándose en dos partes:

1. Una pequeña cantidad de código se incluye en el sector de arranque (*bootstrap*) del *diskette*.
2. El grueso del programa, de 3 Kb, se graba en varios sectores libres del *diskette*. Brain los marca como defectuosos en la tabla de localización de ficheros, para prevenir que se escriba sobre ellos.

Los *diskettes* infectados son fácilmente localizables porque en la etiqueta del volumen del *diskette* aparece la cadena "(c) Brain".

Versiones

Ashar
Similar al virus Clone, con la salvedad de que graba en la etiqueta de volumen del *diskette* la cadena "ASHAR".

Brain-B
Esta versión afecta tanto a los *diskettes* como al disco duro.

Brain-C

Similar al Brain-B, con la salvedad de que no cambia la etiqueta del volumen del disco para no delatarse.

Clone

Esta versión almacena el *copyright* contenido en el sector de arranque original para grabarlo en el sector de arranque infectado, de forma que si se explora dicho sector con un programa de utilidad (por ejemplo, las Utilidades Norton) no sea descubierto.

Nombre: **DOS 62 (UNESCO)**
Tipo: Programa
Sistema afectado: MS-DOS

Historia

Fue descubierto en Moscú en abril de 1988. Se publicó por primera vez en agosto de 1988, cuando se activó en un campamento de verano para niños organizado por la UNESCO.

Descripción

Infecta los programas con extensión .COM. Cuando se ejecuta un programa infectado, el virus infecta otro programa del disco con extensión .COM. Aleatoriamente hace que los programas infectados reinicialicen el sistema al ejecutarlos.

Versiones

62-B
La reinicialización del sistema se ha sustituido por el borrado del programa que se ejecuta.

Nombre: **Dukakis**
Tipo: Programa
Sistema afectado: Macintosh

Historia

Creado como propaganda electoral para las elecciones presidenciales de Estados Unidos.

Descripción

Cuando un sistema es infectado por este virus, si se dan las condiciones necesarias, sacará por pantalla el mensaje "Dukakis for President".

Versiones

No tiene versiones conocidas.

Nombre: **Elk Cloner**
Tipo: Programa
Sistema afectado: Macintosh

Historia

Este virus apareció por primera vez a principios de la década de los ochenta. El virus atacaba a los programas del sistema operativo en los ordenadores de la serie Apple II.

Descripción

Una vez infectado el sistema operativo, el virus se adhiere en los comandos RUN, LOAD, BLOAD y CATALOG. Cuando se utiliza un comando contagiado el virus comprueba, en un acceso al disco, si éste está ya infectado, y si no lo está, procede a contagiarlo.

La acción del virus consistía en imprimir un poema en la pantalla. No se ha observado otro tipo de daños.

Versiones

No tiene versiones conocidas.

Nombre: **Fall (1701, 1704, Cascada)**
Tipo: Programa
Sistema afectado: MS-DOS

Historia

Este virus recibe el nombre de 1701 y 1704 por la cantidad de código, en bytes, que adosa a los programas que infecta.

Descripción

Este virus se adhiere a los ficheros con extensión .COM y .EXE ampliándolos en 1701 y 1704 bytes, respectivamente. Cuando se ejecuta un programa infectado, el virus se hace residente en memoria e infecta cualquier programa con extensiones .COM o .EXE que se ejecute.

Tiene tres características principales:

1. Usa un algoritmo criptográfico que dificulta su detección y el análisis de su código.
2. Contiene un sofisticado algoritmo de activación basado en el tipo de monitor, el tipo de ordenador, la hora y el año.
3. Está diseñado para infectar únicamente ordenadores IBM y clónicos.

El virus sólo se activa en ordenadores con monitores CGA y VGA, en los meses de septiembre, octubre, noviembre o diciembre de los años 1980 ó 1988.

Los efectos de este virus son muy vistosos, ya que produce la paulatina caída de los caracteres de la pantalla, acompañada de algunos ruidos.

Versiones

Fall-B
Esta versión actúa en el otoño de cualquier año.

1704-B

La cascada de caracteres se ha sustituido por una reinicialización del sistema.

1704-C

Similar a la versión 1704-B, excepto que el virus se activa en diciembre de cualquier año.

Nombre: **Flushot-4**
Tipo: Programa
Sistema afectado: MS-DOS

Historia

Al principio de marzo de 1988 apareció en los BBS. Engañaba a los usuarios haciéndoles pensar que era una versión actualizada del Flushot-3, un programa legítimo de protección contra los virus.

Descripción

Las pantallas y menús de este virus son iguales a las del Flushot-3. Sin embargo, cuando se activa limpia la mayoría de los *clusters* importantes del disco duro y modifica la tabla de parámetros de disco (TPD), situada en el sector de arranque, de todos los *diskettes* presentes en el sistema, dejándolos inservibles.

Versiones

No tiene versiones conocidas.

Nombre: **Golden Gate (500)**
Tipo: Sector de arranque
Sistema afectado: MS-DOS

Historia

Es una versión del virus Alameda, que formatea el disco duro cuando el contador interno alcanza un determinado valor.

Descripción

Se activa cuando ha producido 500 infecciones, no realizando ninguna acción antes de este hecho. Cuando se reinicializa el sistema con nuevos *diskettes*, quedan infectados.

Versiones

Golden Gate-B
Se activa cuando ha realizado 30 infecciones.

Golden Gate-C
Esta versión es capaz de infectar discos duros.

Golden Gate-D
En esta versión se ha desactivado el contador interno.

Nombre: **INIT 29**
Tipo: Programa
Sistema afectado: Macintosh

Historia

Los datos que se conocen sobre este virus son bastante confusos.

Descripción

Este virus se activa cuando se ejecuta o selecciona una aplicación infectada. Inicialmente infecta el fichero de sistema y a partir de ahí el virus se adhiere al segmento de apertura de ficheros. Cualquier acción de apertura de ficheros tendrá como consecuencia su infección. El virus no requiere que se ejecute una aplicación para ejercer su acción de copia. Unicamente los ficheros de sistema o las aplicaciones expanden el virus, aunque se puede infectar otro tipo de ficheros.

Cuando se empieza el trabajo con un nuevo disco, el virus intenta contagiarlo. Esta acción produce que aparezca en la pantalla el mensaje "El disco necesita reparaciones menores".

Versiones

No tiene versiones conocidas.

Nombre: **Internet**
Tipo: Gusano
Sistema afectado: Versión 4.3 de UNIX de Berkeley

Historia

El gusano fue desarrollado por un estudiante de la Universidad de Cornell. Afectó a 6000 ordenadores de la red ARPANET el 3 de noviembre de 1988. Atacó indistintamente estaciones de trabajo SUN de la casa Microsystems y máquinas VAX de Digital Equipments que ejecutan la versión 4.3 de UNIX de Berkeley.

Descripción

El programa se infiltró aprovechando un "agujero" en la utilidad del correo electrónico. El gusano se escondía en la memoria mientras creaba un programa que consiguiera de otro ordenador un conjunto de programas. De estos programas extraía los nombres y las cuentas de usuarios válidos, así como sus palabras clave para dirigir nuevos ataques. De esta manera, conseguía acceder a nuevos nodos de la red.

En principio, el programa debía permanecer oculto en el ordenador atacado sin provocar daños. La causa de su descubrimiento estuvo en un error de programación que disparaba el mecanismo de autocopia del gusano, motivo por el cual la red se saturó en cuestión de minutos.

Versiones

No tiene versiones conocidas.

Nombre: **Italian (virus de la pelotita)**
Tipo: Sector de arranque
Sistema afectado: MS-DOS

Historia

Se cree que fue desarrollado en Turín, Italia.

Descripción

Tiene un gran parecido con el Brain, pues divide su código en dos partes, situándose en el sector de arranque y en sectores del área de datos que marca como defectuosos. A diferencia de la primera versión del Brain, afecta tanto a *diskettes* como a discos duros.

Cuando se activa aparece una pelotita que rebota en los bordes de la pantalla y va borrando todos los caracteres que encuentra en su camino. Este virus sólo puede desactivarse apagando el ordenador.

Versiones

No tiene versiones conocidas.

Nombre: **Larry the lounge lizard**
Tipo: Sector de arranque
Sistema afectado: MS-DOS

Historia

Es un juego muy popular entre los usuarios de ordenadores. Su gran difusión se ha debido a las copias piratas.

Descripción

El juego borra todos los ficheros del disco duro cuando se consigue la máxima puntuación.

No hay seguridad de que este programa sea un virus; hay evidencias de que las versiones piratas del juego contienen un código destructivo, por lo que se podría afirmar que es un caballo de Troya.

Versiones

No tiene versiones conocidas.

Nombre: **Mushroom**
Tipo: Gusano
Sistema afectado: Novell Networks MS-DOS

Historia

Fue desarrollado en Australia.

Descripción

Ha sido desarrollado para introducirse en las redes locales Novell mediante *diskettes*. El gusano reproduce la música de un anuncio de desodorante australiano. La poca información disponible impide averiguar el mecanismo que utiliza para viajar a través de la red.

Versiones

No tiene versiones conocidas.

Nombre: **nVIR**
Tipo: Programa
Sistema afectado: Macintosh

Historia

El virus se encontró por primera vez en 1987. La publicación del código original hizo que aparecieran nuevas versiones modificadas. El síntoma que llevó a su descubrimiento fue que el ordenador emitía un sonido cuando se arrancaba una aplicación, pero no siempre que se hacía, ni en todas las aplicaciones.

Descripción

Las investigaciones han demostrado que el virus instala varios servicios del tipo nVIR en la aplicación. También modifica el servicio CODE 0 e instala un servicio INIT 32. Los cambios realizados afectan tanto a aplicaciones como a ficheros de sistema.

A la ya mencionada acción de zumbido hay que añadir como efectos más destacados la pérdida o daño de programas y ficheros de datos, frecuentes caídas del sistema y un mensaje en el sintetizador de voz comunicando que no se tenga miedo.

Versiones

nVIR-A

El virus incorpora un contador que se va decrementando de uno en uno, desde mil, cada vez que se arranca el sistema, y de dos en dos cada vez que se ejecuta un programa infectado. Cuando el contador llega a cero, el ordenador hablará pidiendo calma al usuario, si el sintetizador de voz está conectado; si no lo está, emitirá un zumbido.

nVIR-B

Esta versión no utiliza el sintetizador de voz, simplemente emite un zumbido. Algunas de sus subversiones están programadas para borrar un fichero de forma aleatoria. También varía el número o nombre utilizado en los servicios auxiliares nVIR.

Nombre: **Scores**
Tipo: Programa
Sistema afectado: Macintosh

Historia

Las primeras noticias que se tienen de él datan de finales de 1987 y principios de 1988. Se sospecha que fue desarrollado por un empleado descontento de Electronic Data Systems, EDS. El motivo de la sospecha es que el virus busca especialmente *software* desarrollado por esta empresa y ejerce su función destructiva sobre él.

Descripción

Scores infecta el gestor del sistema, los programas de aplicación, el *note pad* y el fichero Scrapbook. Además crea dos ficheros ocultos de sistema a los que denomina Scores y Desktop.

Después de un período de incubación, el programa empieza su esparcimiento infectando cualquier fichero ejecutable que encuentra. Tras una espera de varios días, el virus busca en el disco cualquier fichero que tenga las marcas utilizadas por los programadores de Macintosh de EDS. Si localiza alguno, modifica su código de manera que si llega a ejecutarse se produce una caída del sistema a los veinticinco minutos de uso. También modifica las actividades de escritura del programa infectado sobre discos, para poder realizar sus cambios pertinentes. Después de siete días entra en acción la última fase de ataque: Quince minutos después de arrancar una de las aplicaciones reseñadas, el virus causará una operación de escritura en un fichero del disco que producirá un error de sistema. Se detecta por la variación del icono del Mac.

Versiones

No tiene versiones conocidas.

Nombre: **Search (DEN ZUK, Venezolano)**
Tipo: Sector de arranque
Sistema afectado: MS-DOS

Historia

Se cree que fue desarrollado en Venezuela.

Descripción

Afecta únicamente a *diskettes* de 5,25 pulgadas de 360 Kb. El virus se hace residente en memoria y no puede ser desactivado pulsando la secuencia de teclas **<Ctrl-Alt-Supr>**. Contiene un fallo que produce intentos de infectar *diskettes* de 3,5 pulgadas. Por esta razón reescribe la tabla de localización de ficheros (FAT) del *diskette*, provocando un fallo de lectura o escritura. No puede infectar discos duros y evita hacerlo para no descubrirse. Si el sistema se reinicializa desde disco duro, el virus se desactiva. Si se reinicializa con un *diskette* no contaminado, éste se infectará.

Hace aparecer en las pantallas CGA, EGA y VGA un gráfico con las palabras "DEN ZUK" después de reinicializar el sistema con la secuencia de teclas **<Ctrl-Alt-Supr>**.

Versiones

Search-B
Es un intento, sin éxito, de eliminar los problemas que tenía la versión original con los *diskettes* de 3,5 pulgadas.

Search-HD
Es capaz de infectar discos duros.

Nombre: **Stoned**
Tipo: Sector de arranque
Sistema afectado: MS-DOS

Historia

El primer lugar donde se dio a conocer fue en Wellington, Nueva Zelanda, a principios de 1988.

Descripción

Como el Brain, se instala en dos partes del *diskette*. Una pequeña parte se graba en el sector de arranque y el grueso del programa en cualquier otra zona del *diskette*.

Cuando se activa aparece en la pantalla el mensaje: "Your computer is now stoned, legalize Marijuana" ("Su ordenador ha sido petrificado, legalicen la marihuana") y causa ciertos daños a la tabla de localización de ficheros (FAT) del *diskette*.

Versiones

Stoned-B

Esta versión es capaz de infectar discos duros. El disco duro se infecta si se arranca el sistema con un *diskette* que porta el virus.

Stoned-C

Similar al Stoned-B. Con esta versión no aparece ningún mensaje por pantalla, por lo que resulta difícil detectarlo.

Nombre: **SYS**
Tipo: Sector de arranque
Sistema afectado: MS-DOS

Historia

Este virus es, en realidad, una modificación del virus Search-HD.

Descripción

El mensaje que aparece en la pantalla se ha sustituido (en este caso no aparece nada) por la desactivación del comando externo del sistema operativo SYS. El comando SYS contaminado actúa de forma normal, accede a los discos en el orden preciso y saca por pantalla el mensaje "sistema transferido" en el momento adecuado, pero en realidad no está haciendo nada.

Versiones

SYS-B
Formatea el disco duro todos los viernes 13 después de 1990. Contiene un fallo que no le permite infectar *diskettes* de 3,5 pulgadas y al intentar hacerlo se descubre.

SYS-C
Produce la reinicialización aleatoria del sistema dos horas después de haberse ejecutado.

Nombre: **Universidad de Lehigh**
Tipo: Programa
Sistema afectado: MS-DOS

Historia

Fue descubierto por primera vez en la Universidad de Lehigh, Bethlehem (Estados Unidos), en 1987. Los operadores de la universidad se dieron cuenta porque la mayor parte de sus discos duros estaban inservibles.

Descripción

El virus se hace residente en memoria cuando el sistema ejecuta el programa COMMAND.COM infectado. Tiene un contador que se incrementa siempre que se hace un acceso a una unidad de disco. Cuando dicho contador toma el valor 5, el virus escribe ceros en los primeros 32 sectores del disco, destruyendo el sector de arranque del directorio raíz y parte del área de datos.

Es fácilmente detectable porque cambia la fecha del fichero COMMAND.COM.

Versiones

No tiene versiones conocidas.

Nombre: **Viena (648)**
Tipo: Programa
Sistema afectado: MS-DOS

Historia

Fue el primer virus publicado enteramente en un libro. Se cree que se desarrolló en Austria, aunque su primera aparición fue en Londres en el otoño de 1988. La longitud del virus es de 648 bytes, por lo que también recibe el nombre de virus 648.

Descripción

El virus se adhiere al principio de los ficheros con extensión .COM. Cuando se ejecuta un programa infectado, el virus busca el siguiente programa con extensión .COM en el directorio de trabajo y lo infecta. Uno de cada ocho programas infectados queda inutilizado, haciendo que el ordenador se bloquee si se intenta ejecutar.

Versiones

Viena-B
Esta versión produce algunas veces errores en los programas infectados, haciendo que no puedan ejecutarse.

Nombre: **Yale**
Tipo: Sector de arranque
Sistema afectado: MS-DOS

Historia

Fue descubierto en Gran Bretaña por Joe Hirst.

Descripción

Se instala en dos partes, una pequeña en el sector de arranque y el resto en el área de datos del disco. El virus permanece residente en memoria incluso después de reinicializar el sistema pulsando la secuencia de teclas **<Ctrl-Alt-Supr>**. La única forma de limpiar la memoria es apagando el ordenador.

Versiones

No tiene versiones conocidas.

Nombre: **2086**
Tipo: Programa
Sistema afectado: MS-DOS

Historia

Este virus trabaja de forma similar al Viernes-13.

Descripción

Al contrario del Viernes-13, este virus sólo infecta una vez los programas con extensión .EXE. Es capaz de infectar el programa COMMAND.COM. Los programas con extensión .COM aumentan su tamaño en 2086 bytes al ser infectados.

Si la fecha del sistema es agosto de 1988 o posterior, cuando se escriben las palabras "Thatcher", "Reagan", "Botha" o "Waldheim", el virus añade a continuación una palabra malsonante diferente para cada uno.

Utiliza la interrupción 21H para detectar si ya está residente en memoria y para infectar los programas .COM y .EXE que se ejecuten.

Versiones

No tiene versiones conocidas.

Nombre: **3066**
Tipo: Programa
Sistema afectado: MS-DOS

Historia

Trabaja de forma similar al virus Fall. Apareció simultáneamente en varias ciudades de Gran Bretaña. También se encontró en Malta.

Descripción

Al ser infectados por este virus, los ficheros .COM y .EXE aumentan su tamaño en 3066 bytes.

El único efecto visible de este virus es la caída de los caracteres que hay en la pantalla, uno a uno, con efectos sonoros. Los caracteres se pueden volver a colocar en su posición pulsando una serie de teclas. No causa mayores daños.

Versiones

No tiene versiones conocidas.

Apéndice D

Programa protector del disco duro

D.1. Introducción

El código que se ofrece a continuación corresponde a un programa de protección lógica contra escritura del disco duro. El lenguaje de programación utilizado es ensamblador correspondiente al compilador Macro Assembler de la casa Microsoft en su versión 5.0.

No se incluye el listado de programas detectores, vacunas y protectores por ser de fácil adquisición, al haber casas de *software* que los distribuyen gratuitamente.

La creciente escalada en las variaciones de los virus y las nuevas creaciones de programas hacen que la mayoría de los programas convencionales "antivirus" se hayan quedado obsoletos. Esto es debido a que el *software* se ha especializado en virus concretos.

Debido a las características del programa, su función es de prevenir y proteger frente a posibles ataques contra el disco duro. Por eso, deberá utilizarse siempre que se vaya a trabajar con *diskettes* que contengan *software* de procedencia dudosa. Este programa puede incluirse en el fichero AUTOEXEC.BAT, con el fin de proporcionar protección del disco duro siempre que se reinicialice el sistema.

D.2. ¿Como hacer ejecutable el programa protector?

El listado deberá escribirse en un editor de líneas o de texto con formato ASCII, con el nombre PROTEC.ASM.

Una vez creado dicho fichero, se procederá siguiendo los pasos que se indican a continuación:

1. Compilar el código fuente con el programa compilador MASM (Macro Assembler de la casa Microsoft). Para ello se debe escribir:

   ```
   MASM PROTEC; <Intro>
   ```

2. Una vez compilado, se obtiene el código objeto contenido en el fichero PROTEC.OBJ. Para obtener el código ejecutable se debe enlazar el programa utilizando el comando externo LINK.EXE del DOS. Para ello se debe escribir:

   ```
   LINK PROTEC; <Intro>
   ```

   En algunas versiones del DOS, al realizar esta operación aparece un mensaje de error referente al segmento de la pila (*stack*), indicando que no encuentra dicho segmento. Este error debe ignorarse, ya que es común cuando se pretende obtener un fichero tipo .COM.
3. Convertir el fichero PROTEC.EXE obtenido en el paso anterior en el fichero PROTEC.COM. Para ello se debe utilizar el comando externo EXE2BIN.EXE del sistema operativo DOS, escribiendo:

   ```
   EXE2BIN PROTEC.EXE PROTEC.COM <Intro>
   ```

D.3. ¿Cómo ejecutarlo?

Para ejecutar el programa, simplemente se tendrá que escribir su nombre en el indicador de comandos (*prompt*) del DOS, con la siguiente secuencia de teclas:

```
PROTEC <Intro>
```

Aparece entonces, por pantalla, un mensaje indicando que el disco duro está protegido contra escritura. Si se quiere desproteger el disco duro, simplemente habrá que

repetir la operación anterior, es decir, volver a escribir la secuencia:

PROTEC <**Intro**>

Cuando se lleva a cabo esta acción aparece un nuevo mensaje por pantalla indicando que el disco duro no está desprotegido contra escritura.

Si el programa ha sido incluido en el AUTOEXEC.BAT, habrá que desactivar previamente el programa protector (tal como se indicó anteriormente) siempre que se vaya a ejecutar un programa de aplicación que realice operaciones de control de entrada/salida sobre el disco duro.

D.4. Listado del programa protector

A continuación se incluye el listado del programa protector:

```
         NAME     PROTEC
         PAGE     60,132
         TITLE    'PROTEC.COM --- protege el disco duro'

CSEG     SEGMENT  PARA PUBLIC 'CODE'
         ORG      100H
         ASSUME   CS:CSEG,DS:CSEG,ES:CSEG,SS:CSEG

PRODIDU  PROC     NEAR

INICIO:  JMP      SHORT VERRES
NINT13:  DW       2 DUP(0000H)
         ADD      [BX],CL
         CMP      AH,05H
         JE       VERUNI
         CMP      AH,03H
         JE       VERUNI
AINT13:  JMP      DWORD PTR CS:[0103H]
VERUNI:  CMP      BYTE PTR CS:[0107H],00H
         JNE      AINT13
         CMP      DL,00H
         JE       AINT13
         CMP      DL,01H
         JE       AINT13
         CMP      DL,03H
```

```
        JE      AINT13
        MOV     AH,03H
        STC
        RETF

VERRES: MOV     DX,0108H
        MOV     AX,CS
        MOV     ES,AX

BUCLE:  DEC     AX
        MOV     DS,AX
        MOV     SI,DX
        MOV     DI,DX
        MOV     CX,0005H
        CLD
        REPZ    CMPSW
        JNE     FINBUC
        CMP     BYTE PTR DS:[0107H],0FH
        JNE     CAMRES
FINBUC: CMP     AX,0001H
        JNE     BUCLE
        MOV     BYTE PTR CS:[0107H],00H
        MOV     AX,3513H
        INT     21H

GINT13: MOV     CS:[0103H],BX
        MOV     CS:[0105H],ES
        PUSH    CS
        POP     DS
        MOV     DX,OFFSET MENSA1
        MOV     AH,09H
        INT     21H

PINT13: MOV     DX,0108H
        MOV     AX,2513H
        INT     21H

SALRES: MOV     DX,0134H
        INT     27H
CAMRES: NOT     BYTE PTR DS:[0107H]
        CMP     BYTE PTR DS:[0107H],00H
        JE      SIPROT
        MOV     DX,OFFSET MENSA2
        JMP     SHORT NOPROT
        NOP
SIPROT: MOV     DX,OFFSET MENSA1
NOPROT: MOV     AH,09H
        PUSH    CS
        POP     DS
        INT     21H
```

```
TERMIN:  INT      20H

PRODIDU  ENDP

MENSA1:  DB '** DISCO PROTEGIDO CONTRA ESCRITURA **$'
MENSA2:  DB '** DISCO NO PROTEGIDO CONTRA ESCRITURA **$'
MENSA3:  DW       0000H
CSEG     ENDS
END      PRODIDU
```

D.5. Breve explicación del programa

En este apartado no se pretende realizar un análisis exhaustivo del programa, simplemente se ofrece una breve explicación de las distintas subrutinas que utiliza.

NINT13	Nueva interrupción 13h creada por el programa.
AINT13	Llamada a la antigua interrupción 13h.
VERUNI	Ver o comprobar la unidad de disco a la que se quiere acceder para realizar operaciones de escritura o formateo.
VERRES	Ver o comprobar si el programa ya está residente en la memoria.
GINT13	Guardar el vector de interrupción de la antigua interrupción 13h.
PINT13	Poner o colocar en la tabla de vectores de interrupción la nueva interrupción 13h.
SALRES	Salir y quedar residente en la memoria.
CAMRES	Cambiar el estado del programa de residente a no residente, y viceversa.
SIPROT	Si se ha protegido el disco duro, aparece el mensaje MENSA1.
NOPROT	Si no se ha protegido el disco duro, es decir, si se levanta la protección, aparece en la pantalla el mensaje MENSA2.
TERMIN	Terminar y devolver el control al DOS.

Apéndice E

Tablas de interrupciones

Tabla E.1. *Interrupciones de la ROM BIOS*

INT	*Función*	*Subfunción*	*Nombre*
00H			División por cero.
01H			Paso a paso.
02H			NMI.
03H			Ruptura.
04H			Desbordamiento.
05H			Impresión de pantalla.
06H			Reservada.
07H			Reservada.
08H			IRQ0 pulsación del temporizador.
09H			IRQ1 de teclado.
0AH			IRQ2 reservada.
0BH			IRQ3 comunicaciones serie (COM2).
0CH			IRQ4 comunicaciones serie (COM1).
0DH			IRQ5 disco duro.
0EH			IRQ6 *diskette*.
0FH			IRQ7 impresora paralelo (LPT1).
10H			Controlador de vídeo.
10H	00H		Definición de modo vídeo.
10H	01H		Definición de tipo de cursor.
10H	02H		Definición de la posición del cursor.
10H	03H		Búsqueda de la posición del cursor.
10H	04H		Búsqueda de la posición del lápiz óptico.
10H	05H		Definición de la página de pantalla.
10H	06H		Inicialización o desplazamiento de la ventana hacia arriba.
10H	07H		Inicialización o desplazamiento de la ventana hacia abajo.

(Continúa)

INT	Función	Subfunción	Nombre
10H	08H		Lectura de un carácter y su atributo en la posición del cursor.
10H	09H		Escritura de un carácter y su atributo en la posición del cursor.
10H	0AH		Escritura de un carácter en la posición del cursor.
10H	0BH		Definición de la paleta, fondo o borde.
10H	0CH		Escritura de un "pixel" gráfico.
10H	0DH		Lectura de un "pixel" gráfico.
10H	0EH		Escritura de un carácter de modo teletipo.
10H	0FH		Búsqueda del modo de vídeo.
10H	10H	00H	Definición del registro de paleta.
10H	10H	01H	Definición del color del borde.
10H	10H	02H	Definición de paleta y del borde.
10H	10H	03H	Intercambio de bits de parpadeo o intensidad.
10H	10H	07H	Búsqueda del registro de paleta.
10H	10H	08H	Búsqueda del color del borde.
10H	10H	09H	Búsqueda de la paleta y del borde.
10H	10H	10H	Definición del registro de color.
10H	10H	12H	Definición del bloque de registros de color.
10H	10H	13H	Definición del estado de color de la página.
10H	10H	15H	Búsqueda del registro de color.
10H	10H	17H	Búsqueda del bloque de registros de color.
10H	10H	1AH	Búsqueda del estado de color de la página.
10H	10H	1BH	Definición de los valores de la escala de grises.
10H	11H	00H y 10H	Carga del tipo de letra del usuario y reprogramación del controlador.
10H	11H	01H y 11H	Carga del tipo de letra de 8 por 14 de la ROM y reprogramación del controlador.
10H	11H	02H y 12H	Carga del tipo de letra de 8 por 8 de la ROM y reprogramación del controlador.
10H	11H	03H	Definición del bloque especificador.
10H	11H	04H y 14H	Carga del tipo de letra de 8 por 16 de la ROM y reprogramación del controlador.
10H	11H	20H	Definición del puntero de tipos de interrupción 1FH.

(Continúa)

INT	Función	Subfunción	Nombre
10H	11H	21H	Preparación de la interrupción 43H para los tipos de usuario.
10H	11H	22H	Preparación de la interrupción 43H para tipos de 8 por 14 de la ROM.
10H	11H	23H	Preparación de la interrupción 43H para tipos de 8 por 8 de la ROM.
10H	11H	24H	Preparación de la interrupción 43H para tipos de 8 por 16 de la ROM.
10H	11H	30H	Búsqueda de información del tipo de letra.
10H	12H	10H	Información sobre configuración.
10H	12H	20H	Selección de impresión de pantalla alternativa.
10H	12H	30H	Definición de líneas de barrido.
10H	12H	31H	Permiso o prohibición de carga de paleta por defecto.
10H	12H	32H	Habilitación o inhabilitación del vídeo.
10H	12H	33H	Habilitación o inhabilitación de la adición de escala de grises.
10H	12H	34H	Habilitación o inhabilitación de la emulación del cursor.
10H	12H	35H	Cambio de la visualización activa.
10H	12H	36H	Permiso o prohibición de refresco de pantalla.
10H	13H		Escritura de una cadena de modo teletipo.
10H	1AH		Búsqueda o definición de los códigos de combinación de visualización.
10H	1BH		Búsqueda de información de funcionalidad o estado.
10H	1CH		Salvaguarda o recuperación del estado de vídeo.
11H			Búsqueda de la configuración del equipo.
12H			Búsqueda del tamaño convencional de memoria.
13H			Controlador de disco.
13H	00H		Reinicialización del sistema de disco.
13H	01H		Búsqueda del estado del sistema de disco.
13H	02H		Lectura del sector.
13H	03H		Escritura del sector.
13H	04H		Verificación del sector.
13H	05H		Formateo de pista.
13H	06H		Formateo de pista defectuosa.

(Continúa)

INT	*Función*	*Subfunción*	*Nombre*
13H	07H		Formateo de la unidad de disco.
13H	08H		Búsqueda de los parámetros de la unidad.
13H	09H		Inicialización de las características del disco duro.
13H	0AH		Lectura completa del sector.
13H	0BH		Escritura completa del sector.
13H	0CH		Posicionamiento.
13H	0DH		Reinicialización del sistema de disco duro.
13H	0EH		Lectura del registro intermedio del sector.
13H	0FH		Escritura del registro intermedio del sector.
13H	10H		Búsqueda del estado de la unidad.
13H	11H		Recalibrado de la unidad.
13H	12H		Diagnóstico del controlador de la RAM.
13H	13H		Diagnóstico del controlador de la unidad.
13H	14H		Diagnóstico interno del controlador.
13H	15H		Búsqueda del tipo de disco.
13H	16H		Búsqueda del estado de cambio de disco.
13H	17H		Definición del tipo de disco.
13H	18H		Definición del tipo de medio para formatear.
13H	19H		Aparcamiento de cabezas.
13H	1AH		Formateo de la unidad ESDI.
14H			Controlador del puerto de comunicaciones serie.
14H	00H		Preparación del puerto de comunicaciones.
14H	01H		Envío de un carácter al puerto de comunicaciones.
14H	02H		Lectura del carácter del puerto de comunicaciones.
14H	03H		Estado del puerto de comunicaciones.
14H	04H		Preparación completa del puerto de comunicaciones.
14H	05H		Control completo del puerto de comunicaciones.
15H			Extensiones de entrada/salida.
15H	00H		Arranque del motor del casete.
15H	01H		Parada del motor del casete.
15H	02H		Lectura del casete.

(Continúa)

INT	Función	Subfunción	Nombre
15H	03H		Grabación en el casete.
15H	0FH		Interrupción periódica del formateo de la unidad ESDI.
15H	21H	00H	Lectura de información sobre errores detectados en la prueba de arranque (POST).
15H	21H	01H	Escritura de los errores detectados en la prueba de arranque (POST).
15H	4FH		Detección del código del teclado.
15H	80H		Adquisición del control de dispositivo.
15H	81H		Abandono del control de dispositivo.
15H	82H		Fin del proceso.
15H	83H		Espera de acción.
15H	84H		Lectura del mando de *joystick*.
15H	85H		Tecla <**Pet Sis**>.
15H	86H		Retardo.
15H	87H		Movimiento de bloques de datos con la memoria extendida.
15H	88H		Tamaño de la memoria extendida.
15H	89H		Paso a modo protegido.
15H	90H		Espera de dispositivo.
15H	91H		Prueba tras arranque del dispositivo (POST).
15H	C0H		Identificación del equipo.
15H	C1H		Búsqueda de la dirección del área de datos extendida del BIOS.
15H	C2H	00H	Habilitación o inhabilitación del dispositivo puntero.
15H	C2H	01H	Reinicialización del dispositivo puntero.
15H	C2H	02H	Definición de la tasa de muestreo.
15H	C2H	03H	Definición de la resolución.
15H	C2H	04H	Identificación del tipo de dispositivo puntero.
15H	C2H	05H	Inicialización del interfaz del dispositivo puntero.
15H	C2H	06H	Definición de la escala o búsqueda del estado.
15H	C2H	07H	Definición de la dirección del programa de tratamiento del dispositivo puntero.
15H	C3H		Definición del tiempo de espera.
15H	C4H		Selección de la opción programable.
16H			Controlador del teclado.
16H	00H		Lectura del carácter desde el teclado.
16H	01H		Búsqueda del estado del teclado.

(Continúa)

INT	Función	Subfunción	Nombre
16H	02H		Búsqueda de las marcas del teclado.
16H	03H		Definición de la tasa de repetición.
16H	04H		Definición de la pulsación del teclado.
16H	05H		Colocación del carácter y búsqueda del código.
16H	10H		Lectura del carácter del teclado expandido.
16H	11H		Búsqueda del estado del teclado expandido.
16H	12H		Búsqueda de las marcas del teclado expandido.
17H			Controlador del puerto paralelo de impresora.
17H	00H		Envío del carácter a la impresora.
17H	01H		Inicialización del puerto de impresora.
17H	02H		Búsqueda del estado de la impresora.
18H			ROM del BASIC.
19H			Reinicialización (*reboot*) del sistema.
1AH			Controlador del reloj de tiempo real (CMOS).
1AH	00H		Búsqueda del contador de pulsaciones de reloj.
1AH	01H		Definición del valor del contador de pulsaciones.
1AH	02H		Búsqueda de la hora.
1AH	03H		Definición de la hora.
1AH	04H		Búsqueda de la fecha.
1AH	05H		Fijación de la fecha.
1AH	06H		Conexión de la alarma.
1AH	07H		Desconexión de la alarma.
1AH	0AH		Búsqueda del contador de días.
1AH	0BH		Definición del contador de días.
1AH	80H		Definición de la fuente de sonido.

Tabla E.2. *Interrupciones del DOS*

INT	Función	Nombre
21H	00H	Fin del proceso.
21H	01H	Entrada de un carácter con eco.
21H	02H	Salida de un carácter.
21H	03H	Entrada auxiliar.
21H	04H	Salida auxiliar.
21H	05H	Salida a impresora.

(Continúa)

INT	Función	Nombre
21H	06H	E/S directa a consola.
21H	07H	Entrada directa de un carácter sin eco.
21H	08H	Entrada de caracteres sin eco.
21H	09H	Visualización de una cadena de caracteres.
21H	0AH	Entrada desde el teclado.
21H	0BH	Comprobación del estado de entrada.
21H	0CH	Borrado del registro de entrada y lectura de datos.
21H	0DH	Borrado del disco.
21H	0EH	Selección del disco.
21H	0FH	Apertura de un fichero.
21H	10H	Cierre de un fichero.
21H	11H	Búsqueda del primer fichero.
21H	12H	Búsqueda del siguiente fichero.
21H	13H	Borrado de un fichero.
21H	14H	Lectura secuencial.
21H	15H	Escritura secuencial.
21H	16H	Creación de un fichero.
21H	17H	Cambio del nombre.
21H	18H	Reservada.
21H	19H	Búsqueda del disco actual.
21H	1AH	Colocación de la dirección del área de transferencia del disco.
21H	1BH	Búsqueda del disco por defecto.
21H	1CH	Búsqueda del disco.
21H	1DH	Reservada.
21H	1EH	Reservada.
21H	1FH	Reservada.
21H	20H	Reservada.
21H	21H	Lectura aleatoria.
21H	22H	Escritura aleatoria.
21H	23H	Cálculo del tamaño del fichero.
21H	24H	Asignación del número relativo de registro.
21H	25H	Asignación del vector de interrupciones.
21H	26H	Creación de un PSP nuevo.
21H	27H	Lectura aleatoria de un bloque.
21H	28H	Escritura aleatoria de un bloque.
21H	29H	Alteración del nombre del fichero.
21H	2AH	Búsqueda de la fecha.
21H	2BH	Asignación de la fecha.
21H	2CH	Búsqueda de la hora.
21H	2DH	Asignación de la hora.
21H	2EH	Asignación de la bandera de verificación.

(Continúa)

INT	Función	Nombre
21H	2FH	Cálculo de la dirección del área de transferencia del disco.
21H	30H	Búsqueda del número de versión MS-DOS.
21H	31H	Fin y paso a residente.
21H	32H	Reservada.
21H	33H	Asignación de la señal de interrupción.
21H	34H	Reservada.
21H	35H	Búsqueda del vector de interrupción.
21H	36H	Búsqueda de información sobre la situación del disco.
21H	37H	Reservada.
21H	38H	Búsqueda o asignación de información sobre el país.
21H	39H	Creación de un directorio.
21H	3AH	Borrado de un directorio.
21H	3BH	Definición del directorio actual.
21H	3CH	Creación de fichero.
21H	3DH	Apertura de fichero.
21H	3EH	Cierre de fichero.
21H	3FH	Lectura del fichero o dispositivo.
21H	40H	Escritura en el fichero o dispositivo.
21H	41H	Borrado de fichero.
21H	42H	Asignación del puntero de fichero.
21H	43H	Búsqueda o asignación de atributos.
21H	44H	Control de E/S (IOCTL).
21H	45H	Duplicación.
21H	46H	Redireccionamiento.
21H	47H	Búsqueda del directorio actual.
21H	48H	Colocación del bloque de memoria.
21H	49H	Liberación de un bloque de memoria.
21H	4AH	Redefinición del tamaño de memoria.
21H	4BH	Ejecución del programa (EXEC).
21H	4CH	Fin del proceso con el código de retorno.
21H	4DH	Espera del código de retorno.
21H	4EH	Búsqueda del primer fichero.
21H	4FH	Búsqueda del siguiente fichero.
21H	50H	Reservada.
21H	51H	Reservada.
21H	52H	Reservada.
21H	53H	Reservada.
21H	54H	Búsqueda del indicador de verificación.
21H	55H	Reservada.

(Continúa)

INT	Función	Nombre
21H	56H	Cambio del nombre del fichero.
21H	57H	Búsqueda o definición de la fecha y la hora del fichero.
21H	58H	Búsqueda o definición de la estrategia de colocación.
21H	59H	Búsqueda de información extensa sobre ficheros.
21H	5AH	Creación de un fichero temporal.
21H	5BH	Creación de un fichero nuevo.
21H	5CH	Bloqueo o desbloqueo de una región de un fichero.
21H	5DH	Reservada.
21H	5EH	Búsqueda del nombre del ordenador, búsqueda o definición del arranque de la impresora.
21H	5FH	Redireccionamiento de los periféricos.
21H	60H	Reservada.
21H	61H	Reservada.
21H	62H	Búsqueda de la dirección de PSP.
21H	63H	Búsqueda de la tabla de bytes.
21H	64H	Reservada.
21H	65H	Búsqueda de información completa sobre el país.
21H	66H	Asignación del código de página.
21H	67H	Definición del contador de control.
21H	68H	Volcado de un fichero.
22H		Tratamiento de finalización.
23H		Dirección de tratamiento de <**Ctrl-C**>.
24H		Tratamiento de errores críticos.
25H		Lectura directa de disco.
26H		Escritura directa en disco.
27H		Fin quedando residente.
28H		Reservada.
29H		Reservada.
2AH		Reservada.
2BH		Reservada.
2CH		Reservada.
2DH		Reservada.
2EH		Reservada.
2FH		Interrupción múltiple.
2FH	01H	Impresión de memoria tampón.

Indice alfabético

A

B

C

D

E

J

L

M

N

O

P

R

S


PREMIO PC MAGAZINE
MEJOR UTILIDAD 1991


PREMIO PC MAGAZINE
MEJOR UTILIDAD 1993



RECOMENDACION PC WORLD
MEJOR PRODUCTO ANTIVIRUS



RECOMENDACION PC MAGAZINE
MEJOR PRODUCTO ANTIVIRUS


CATACOM 93
MEJOR ANTIVIRUS



NUEVA VERSI

Incorpora los últimos virus aparec

CLUB HELPVIRUS

Servicio permanente de actualización

ANTIVIRUS ANYWARE



Protección Detección y Eliminaci
de virus informáticos para IBM PC y compatib

Protección permanente frente a virus conocidos y desconocidos

Detección y eliminación de virus en ficheros infectados

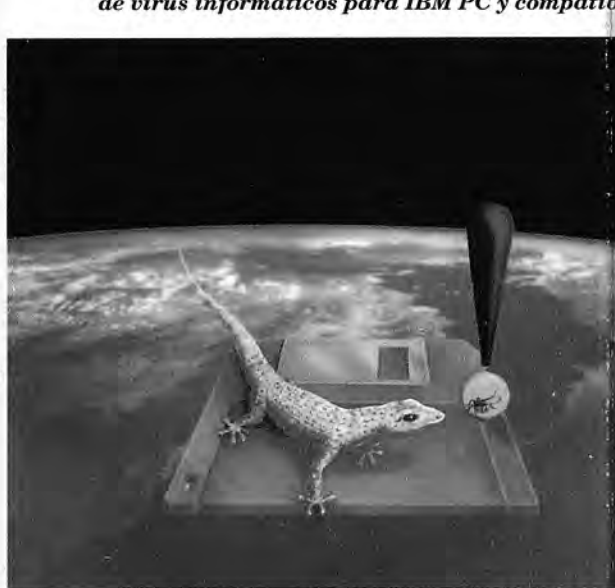
Analiza unidades y ficheros comprimidos

Inmunización de ficheros

Uso por menú o línea de comandos

Compatibilidad con Windows y redes de área local

Contempla el 100% de los virus españoles



Comentarios de las revistas especializadas

PC WORLD
ESPAÑA

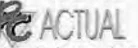
"Si usted quiere el mejor sistema antivirus del mercado, el paquete de Anyware es ideal."
Septiembre 1991

PC MAGAZINE

"Excelente precio para un producto de inmejorable calidad."
Diciembre 1991



"La principal virtud del Antivirus Anyware es su elevada eficacia a la hora de detectar y eliminar virus conocidos." "Anyware mantiene el producto actualizado con gran rapidez, proporcionando una elevada protección de los datos vitales."
Marzo 1992



"La solución antivirus de Anyware destaca por su alta capacidad detectora y por su interface de usuario."
Diciembre 1992

PC WORLD
ESPAÑA

"Antivirus Anyware es la mejor opci protección frente a los actuales,...,destaca por su soporte técn grado de actualización..."
Mayo 1



"Tal vez el hecho de que en nuestros de test hubiera bastantes de los habi en España..., sea el responsable certera detección de Anyware: un 100%
Julio

Si quiere estar seguro llame a Anyware



(91) 556 92 16

Anyware Seguridad Informatica, S.A.
Orense, 36. 3ª Planta 28020 Madrid

